

Decoding ARISSat-1



Beginners - How to Begin? (Part III)

in this issue

AMSAT Announcements2

Apogee View3

by Barry Baines • WD4ASW

Getting Started on the4

Amateur Radio Satellites (Part III)

by Keith Baker • KB1SF/VA3KSF

Introducing AMSAT Fox8

by Anthony Monterio • AA2TX

More on Converting California 11

Amplifier 31732 Downconverters

for 2.4 GHz

by John M. Franke • WA4WDL

John Champa, K8OCL, SK13

by Dee Interdonato • NB2F

El Cheapo RX Antenna for 14

AO-7 Mode A Operation

by John Neeley • W6ZKH

The BPSK1000 Telemetry 16

Modem for ARISSat-1

by Phil Karn • KA9Q

ARISSatTLM - A Software27

Demodulator and Telemetry

Display Program for the ARISSat-1

Satellite

by Douglas D. Quagliana • KA2UPW/5

How You Can Help Build New and Exciting Satellites

Donate to the President's Club

Gold, Silver, Bronze and Core levels are available to match your ability to participate.

Cash Gifts

Visa, or MasterCard or checks are accepted. And, you can specify how your contribution is to be used.

Gift of Life Insurance

US taxpayers may be able to receive a significant income tax deduction by making The Radio Amateur Satellite Corporation the owner and beneficiary of life insurance policies.

Gift of Stocks or other Securities

US taxpayers should be able to avoid capital gains taxes on appreciated securities and receive a deduction for their fair market value.

Bequest

A codicil in your will, naming The Radio Amateur Satellite Corporation as a beneficiary will help insure the continuance of the Amateur Radio Satellite program.

Call the AMSAT-NA office at 301-589-6062 for questions on any or all of these ways you can help build new and exciting satellites.

Support AMSAT-NA

AMSAT Announcements

WAS-Satellite Certificate Now Accepted as Credit for AMSAT Awards

AMSAT Director of Contests and Awards Bruce Paige, KK5DO, says that in some cases he can credit holders of the ARRL Worked All States (with the Satellite endorsement) toward appropriate AMSAT Satellite Operating Awards.

He will accept a scanned copy of your Worked All States-Satellite certificate as proof of 50 of the 60 contacts to fill in the missing states for the AMSAT Sexagesimal Award or to qualify for the AMSAT Communication Achievement Award. Contact Bruce via e-mail with your specific details to see if you qualify (kk5do@amsat.org).

Unfortunately, the ARRL W.A.S. certificate cannot be used towards the South Africa AMSAT Award. As all contacts for this award must be done on LEO satellites, there is no way to determine if LEO or HEO was used on the certificate.

Full information on all of the AMSAT Satellite Operating Awards can be found on-line at: <http://www.amsat.org/amsat-new/awards/> 🌐

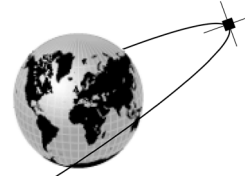
AMSAT's Mission

AMSAT is a non-profit volunteer organization which designs, builds and operates experimental satellites and promotes space education. We work in partnership with government, industry, educational institutions and fellow Amateur Radio societies. We encourage technical and scientific innovation, and promote the training and development of skilled satellite and ground system designers and operators.

AMSAT's Vision

Our Vision is to deploy satellite systems with the goal of providing wide-area and continuous coverage. AMSAT will continue active participation in human space missions and support a stream of LEO satellites developed in cooperation with the educational community and other amateur satellite groups.





Radio Amateur Satellite Corporation (AMSAT-NA)
850 Sligo Ave., Silver Spring, MD 20910-4703
Telephone: 301-589-6062 – Toll Free: 888-322-6728
Facsimile: 301-608-3410
AMSAT-NA Club Callsign: W3ZM
AMSAT-NA Web site: <http://www.amsat.org>

The AMSAT Journal Staff

Editor-in-Chief: Ed Long, WA4SWJ, wa4swj@amsat.org
Assistant Editor: Bill Hook, W3QBC, w3qbc@amsat.org
Assistant Editor: JoAnne Maenpaa, K9JKM, k9jkm@amsat.org
Circulation: Martha Saragovitz, martha@amsat.org

AMSAT-NA Board of Directors

Barry Baines, WD4ASW, wd4asw@amsat.org
Alan Biddle, WA4SCA, wa4sca@amsat.org
Tom Clark, K3IO, k3io@amsat.org
Drew Glasbrenner, KO4MA, ko4ma@amsat.org
Lou McFadin, W5DID, w5did@amsat.org
Tony Monteiro, AA2TX, aa2tx@amsat.org
Gould Smith, WA4SXM, wa4sxm@amsat.org

AMSAT-NA Senior Officers

President: Barry Baines, WD4ASW
Executive Vice-President: Open
Treasurer: Keith Baker, KB1SF/VA3KSF
Secretary: Alan Biddle, WA4SCA
Manager: Martha Saragovitz
Vice-President, Engineering: Tony Monteiro, AA2TX
Vice-President, Operations: Drew Glasbrenner, KO4MA
Vice-President User Services: Gould Smith, WA4SXM
Vice-President Human Spaceflight: Open
Vice President, Marketing: Open

Honorary Positions

Immediate Past President: Rick Hambly, W2GPS
President Emeritus: Tom Clark, K3IO
Founding President: Perry Klein, W3PK

Editorial Office: Ed Long, WA4SWJ, 4420 SE 106TH ST, Belleview, FL, 34420-6835. Please e-mail Journal submissions to: journal@amsat.org.
Editor's telephone: 262-374-1349 (cell). **Advertising Office:** AMSAT-NA Headquarters, 850 Sligo Avenue, Suite 600, Silver Spring, MD 20910-4703. Telephone: 301-589-6062.

The AMSAT Journal (ISSN: 1407-3076) is published bi-monthly (Jan/Feb, Mar/Apr, May/Jun, Jul/Aug, Sep/Oct, Nov/Dec) by AMSAT-NA, 850 Sligo Avenue, Suite 600, Silver Spring, MD 20910-4703. Telephone: 301-589-6062, fax: 301-608-3410. Periodicals postage paid at Silver Spring, MD and additional mailing offices.

Postmaster: Send address changes to *The AMSAT Journal*, 850 Sligo Avenue, Suite 600, Silver Spring, MD 20910-4703.

Opinions expressed in *The AMSAT Journal* are those of the article authors and are not necessarily those of AMSAT-NA. Copyright © 2010 by AMSAT-NA. The Radio Amateur Satellite Corporation. AMSAT is a registered trademark. Reproduction of material from *The AMSAT Journal* by mechanical, electronic, photocopy or other means is prohibited unless written permission is obtained from the author.

The AMSAT Journal staff is always interested in article submissions. Whenever possible, submissions should be sent via e-mail to journal@amsat.org using plain text or word processor files; photos or figures in TIF, GIF or JPEG formats. Kindly do not embed graphics or photos in your manuscript - we prefer receiving those as separate files. AMSAT-NA reserves the right to select material for *The AMSAT Journal* based on suitability of content and space considerations. The editors of this publication are volunteers giving freely of their talents, time and efforts to produce *The AMSAT Journal*.

When we are focused on moving forward, it can be difficult to stop for a moment to reflect upon the accomplishments made over the past year. After all, we're 'wired' to look at the future and not sit back and presume that what has been accomplished to date is sufficient to keep an organization vibrant and growing.

As I'm writing this note just before Christmas, three things stick out as immediate reminders of how much AMSAT has accomplished:

1. ARISSat-1/RadioSkaf-V along with a backup unit is in Moscow. AMSAT committed to delivering an 'Amateur Radio Experiment' for deployment during EVA#28 in February 2011. The two units were shipped to NASA's Johnson Space Center in early October for furtherance to RSC-Energia. AMSAT indeed has fulfilled its pledge to RSC-Energia and ARISS-International.
2. December 21, 2010 is the 10th Anniversary of the first ARISS school contact. Bill Shepherd, KDGSL conducted the first contact during Expedition 1 with Burbank School in Burbank, IL. Since that first contact, every Expedition Team has conducted ARISS school contacts, totaling over 560 schools or organizations. This is a significant accomplishment and reflects the commitment and dedication of ARISS volunteers, many of whom are AMSAT members who initiated Amateur Radio in manned space with SAREX. AMSAT has served with distinction as the center of technical development for ARISS, including placement of Amateur Radio equipment on the ISS, the development of SuitSat-1 and now the delivery of ARISSat-1.
3. In early December AMSAT sent a fund raising letter to our membership asking for their financial support of AMSAT. To date, more than \$37,000.00 has been contributed in response to that letter. This includes, over \$1,200 donated through Internet resources, such as PayPal via Facebook postings and Web page 'widgets'. A widget was placed on the AMSAT homepage. This new approach toward encouraging donations from people, many who are not AMSAT members, is very encouraging.

If you haven't donated to our fundraising campaign, please consider doing so. Your

financial support is critical as explained in the fund raising letter that each AMSAT member received.

Looking forward, here are several items of note:

ARISSat-1 will be deployed in 2011. That said, there is always something happening outside of AMSAT that impacts us. ARISSat-1 sat for over a month at Johnson Space Center, Houston awaiting customs paperwork from RSC-Energia that was needed before shipment could take place. When that paperwork was received, the shipment was made, anticipating that it would be released from Russian Customs within a few days once it arrived in Moscow. The shipment arrived in Moscow on Wednesday, 8 December after it was delayed in transit for several days on its trip from Houston to Moscow. Issues were then raised regarding the paperwork prepared by RSC-Energia that have prevented ARISSat-1 from being released by Russian Customs. As I write this, the satellite is apparently still in Customs. While there is hope that ARISSat-1 will be released from Customs 'shortly', there is the possibility that the process could take several more weeks, seriously disrupting the entire schedule for deployment.

Meanwhile, earlier this Fall, AMSAT arranged for Lou McFadin, W5DID, to fly to Moscow to assist with the integration and testing of ARISSat-1 at RSC-Energia. Mark Steiner, W3MS, accompanied Lou as a NASA representative. Due to the lead-time for Russian visas, we had to specify in October the timeframe for the visit; no extensions would be allowed once they arrived in Russia. We arranged for the period of 3-22 December for the visas with the expectation that testing would be completed before Christmas. In addition, with the flight of the Expedition 27 crew on a Soyuz in mid-December, there were 'blackout dates' as to when Mark and Lou could fly to/from Russia given that NASA's first priority was the Expedition 27 deployment. The celebration of Christmas in Russia is under the Russian Orthodox calendar. This means that RSC-Energia shuts down from 31 Dec 10 to 10 Jan 11. For ARISSat-1 to make the Progress 41P launch at the end of January, it

continued on page 23 ...



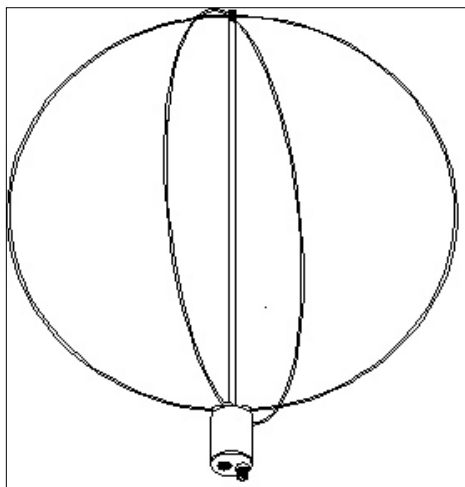
Getting Started on the Amateur Radio Satellites (Part III)

by Keith Baker, KB1SF/VA3KSF, kb1sf@amsat.org

(The bulk of this article was previously published as "Working Your First Amateur Radio Satellite (Part III)" in the May 2010 issue of *Monitoring Times*, Brasstown, NC 28902)

I trust by now a number of you are up and running on our FM birds and are having fun collecting new grid squares or working DX with this (for you) newfound part of our wonderful hobby. However, my hunch is that your arm is probably getting tired while working these satellites using just a small, portable, handheld radio and a handheld Yagi of some sort.

In anticipation of springtime's warmer weather (in the Northern Hemisphere at least!) my guess is that you'd like to now begin investigating a more permanent antenna array for your satellite station. For beginners on a budget, I suggest you consider some form of omnidirectional antenna.



The Eggbeater is a good omnidirectional base station antenna useful for working the LEO birds. (Courtesy: M2 Antennas)

That's because their use tremendously simplifies building your satellite station, as no rotors, cross booms, or rotor interfaces are needed. Use of omnidirectional antennas also greatly simplifies the satellite tracking part of this activity as it will allow you to fully concentrate on trying to hear, find (and/or track) your own downlink signals while working the bird as it rapidly moves across the sky.

But, unfortunately, and as we have discussed, not all omnidirectional antennas are suitable for satellite work. So, in this edition of our beginner's corner, I'll once again offer some tips to help you optimize your base station antennas for the satellites.

More Satellite Antenna Considerations

Contrary to what you might have heard (from well meaning veteran satellite ops) that only a cross-polarized set of multi-element Yagi antennas mounted on a non-metallic cross boom will do, I know from my own personal experiences that such talk is largely bunkum. That is, just as with most other pursuits in Amateur Radio, while the "ultimate" satellite base station antenna array may sport one or more circularly polarized Yagi antennas all mounted on a fiberglass cross boom and turned by an (expensive!) commercial alt-azimuth rotator, you can usually still get excellent results on the LEO birds for a whole lot less time, money and effort.

If you already have a VHF and UHF base station set up for scanning or for use on the amateur bands, you probably also have an external VHF or UHF antenna of some sort connected to it. Unfortunately, the gain of most of these terrestrial antennas occurs at the point in a satellite's orbit where it is farthest away from your station (at the horizon) and its downlink signal is at its weakest.

What's more, as the satellite rises above your horizon, it will gradually move outside the beam width of most terrestrially optimized antennas to the point that, when it is at its closest approach to you (directly overhead) you may not hear the satellite ... and it may not hear you ... at all!

Remember, too, that Amateur Radio satellites are both tumbling and spinning in space. As we discussed in previous columns, cross-polarizing linear antennas results in a *huge* loss of gain. This means that if the antenna on the satellite is horizontally polarized and your antenna on the Earth is vertically polarized (or vice versa), you may not receive much of anything on the ground, no matter how much power is being transmitted to or from the satellite.

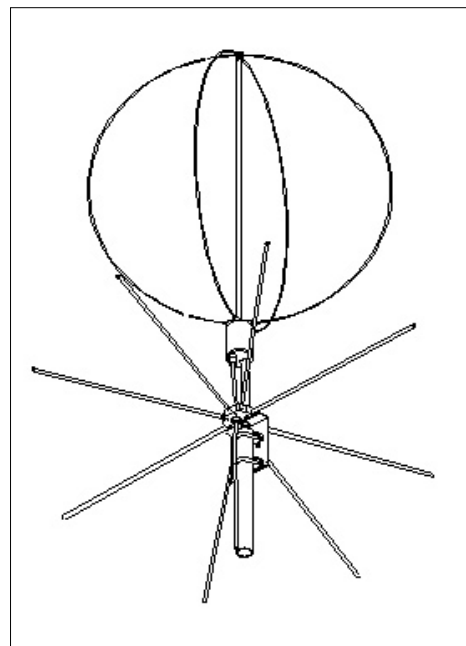
To help minimize these problems, satellite builders usually incorporate what are called circularly polarized antennas into their satellites. Building circularly polarized antennas into a satellite helps minimize the effects of antenna cross-polarization losses on the ground as the satellite moves through space. That's because the difference between

right-hand circular polarization and left-hand circular polarization is only about 3 dB.

Thankfully, there are a couple of relatively simple, omnidirectional antennas that are also specifically designed to achieve this high angle, circular signal polarization pattern without *also* costing you a fortune ... or making your home look like a NASA tracking station!

Scrambled Eggs, Anyone?

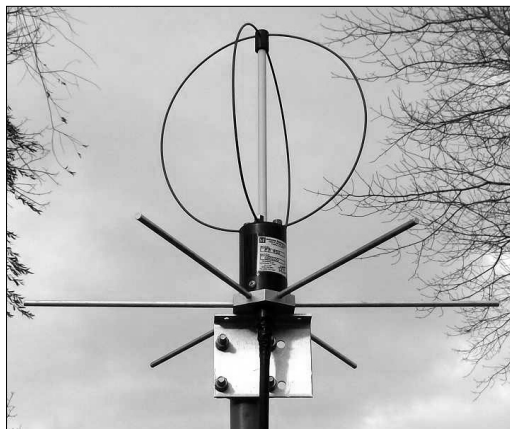
One relatively inexpensive omnidirectional base station antenna that is useful for LEO satellite work is called an Eggbeater. The Eggbeater antenna looks a lot like its namesake ... an ordinary kitchen eggbeater. It's composed of two full-wave loops of wire (or some other rigid metal material) fed 90 degrees out of phase with each other. Some designs even sport parasitic reflector elements underneath the array to give the antenna more elevated gain. At the horizon, the eggbeater exhibits a



Adding ground plane elements under the Eggbeater increases the overall upward gain of the antenna. (Courtesy: M2 Antennas)

horizontally polarized linear pattern, which also makes it useful for weak signal VHF or UHF terrestrial work. However, at higher elevations, the antenna exhibits an ever more right-hand circular radiation pattern, which makes it *ideal* for satellite work.

Gerald Brown, K5OE, has published an excellent Web article on how to home-brew



This photo shows a view of the author's 70cm, M2 Eggbeater antenna. Note the ground radials mounted underneath the phasing coil. (Courtesy: KB1SF)

satellite-optimized eggbeater antennas at: <http://victrolla.homeip.net/wo5s/junkpile/432/eggbeater2.pdf>. Eggbeaters are also available from commercial antenna manufacturers such as M2 Antennas of Fresno, California (<http://www.m2inc.com>).

I'm currently using a pair of commercially made M2 eggbeater antennas at my home QTH and find they work reasonably well for the LEO birds, particularly if the antenna's reflector elements are installed.

Now, granted, eggbeaters *won't* give you horizon-to-horizon, S9 signals either into (or out of) the birds like a long-boom Yagi will. But, remember, satellite work is by default, weak signal work. Using a reasonable amount of uplink power (usually 25 watts or less) and a pre-amplifier for the downlink mounted either on the antenna mast (or if your high quality feedline is short enough) mounted in the shack, I've found I can get good results with these antennas on most overhead (or near overhead) passes. As a general rule, I don't bother trying to work a satellite pass with my eggbeaters unless the satellite will be at least 45 to 50 degrees above my horizon at its highest point.

A Quadrifilar WHAT?

Another omnidirectional antenna design suitable for satellite work is a Quadrifilar Helix (or Quadrafilair Helicoidal) antenna. A Quadrafilair Helix antenna consists of four quarter-wavelength or half-wavelength elements fed with a 90-degree phase difference. The polarization is circular, and the beam widths are often greater than 90 degrees, which means this antenna will cover a HUGE chunk of the sky.

These antennas are also relatively small

and fairly easy to build out of common materials such as copper tubing and PVC pipe. However, element lengths and spacing have to be *very* precise to achieve a truly circularly polarized pattern.

A number of ham operators (and others who are also interested in weather satellite reception) offer design tips and construction details for these antennas via various Web sites including http://www.n8imo.com/qha_4.html and http://perso.wanadoo.es/dimoni/ant_qha.htm. Yet another Web site on the subject, (<http://www.jcoppens.com/ant/qfh/calc.en.php>) sports a helpful online calculator where element lengths and spacing for these antennas can be calculated simply by entering the desired resonant frequency.

The Lindenblad

Yet another omnidirectional antenna design that can be useful for satellite work is the Lindenblad. The antenna is named for Nils Lindenblad of the Radio Corporation of America (RCA) who, back in the early 1940s, began experimenting with antenna designs that might be useful for the (then emerging) television broadcast industry. The antenna uses four dipoles spaced equally around a 1/3-wavelength circle with each element canted at a 30-degree angle from horizontal.

Like the Quadrafilair Helix, construction articles on how to roll your own Lindenblad abound on the Internet. Howard Sodja, W6SHP, optimized the Lindenblad design for satellite work in a series of articles for *The AMSAT Journal* in the early 1990s. The articles can still be found in the AMSAT Web archives at: <http://www.amsat.org/amsat/articles/w6shp/lindy.html>. In addition, AMSAT's current Vice President of Engineering and BoD Member, Tony Monteiro, AA2TX, has written extensively on the Lindenblad design. Construction details of his 70 cm version of the Lindenblad appeared in the Proceedings for the 2006 AMSAT Annual Meeting and Space Symposium at: <http://www.qsl.net/nwlnarn/sat/70ParaLindy.pdf>.

Directional Antennas

As the name implies, directional antennas focus RF energy in one direction. Not only do these antennas allow you to transmit your signal to satellites that are farther away from you, they also help your ground station pick up weaker signals, provided that the antennas are pointed in the right direction. As all satellite work is weak signal work,



Here's a partial view of the phasing coil and ground plane of the author's 2 m, M2 Eggbeater antenna. Note that the ground plane radials are offset by about 1/2 wavelength (10.5 inches) from the top of the phasing coil and driven elements. (Courtesy: KB1SF)

anything that boosts an already weak satellite downlink signal is a good thing.

Many satellite operators use some form of Yagi antenna in their Earth stations. The design is named for its Japanese inventor, Yagi-Uda, and consists of one or more dipoles that are fed with RF and act as driven elements. Parasitic (that is, non-fed) elements (called reflectors) are then mounted in back of the driven element and one or more parasitic elements (called directors) are mounted in front. The whole array is then mounted on a cross boom of some sort.

Yagi antennas can be either linear or circularly polarized. Yagis with only one row of elements are linearly polarized (either horizontal or vertical depending on which way you mount them). However, Yagis with two rows offset by a 90-degree phase difference are circularly polarized (either right-hand or left-hand, looking down the antenna from the rear).

As I've discussed, for satellite communication, circular polarization is desirable because the difference in loss between right-hand (RHCP) and left-hand (LHCP) circular polarization is only about 3 dB. And while this loss represents about half of your uplink or downlink signal, remember that the difference between horizontal and vertical polarization is theoretically infinite.



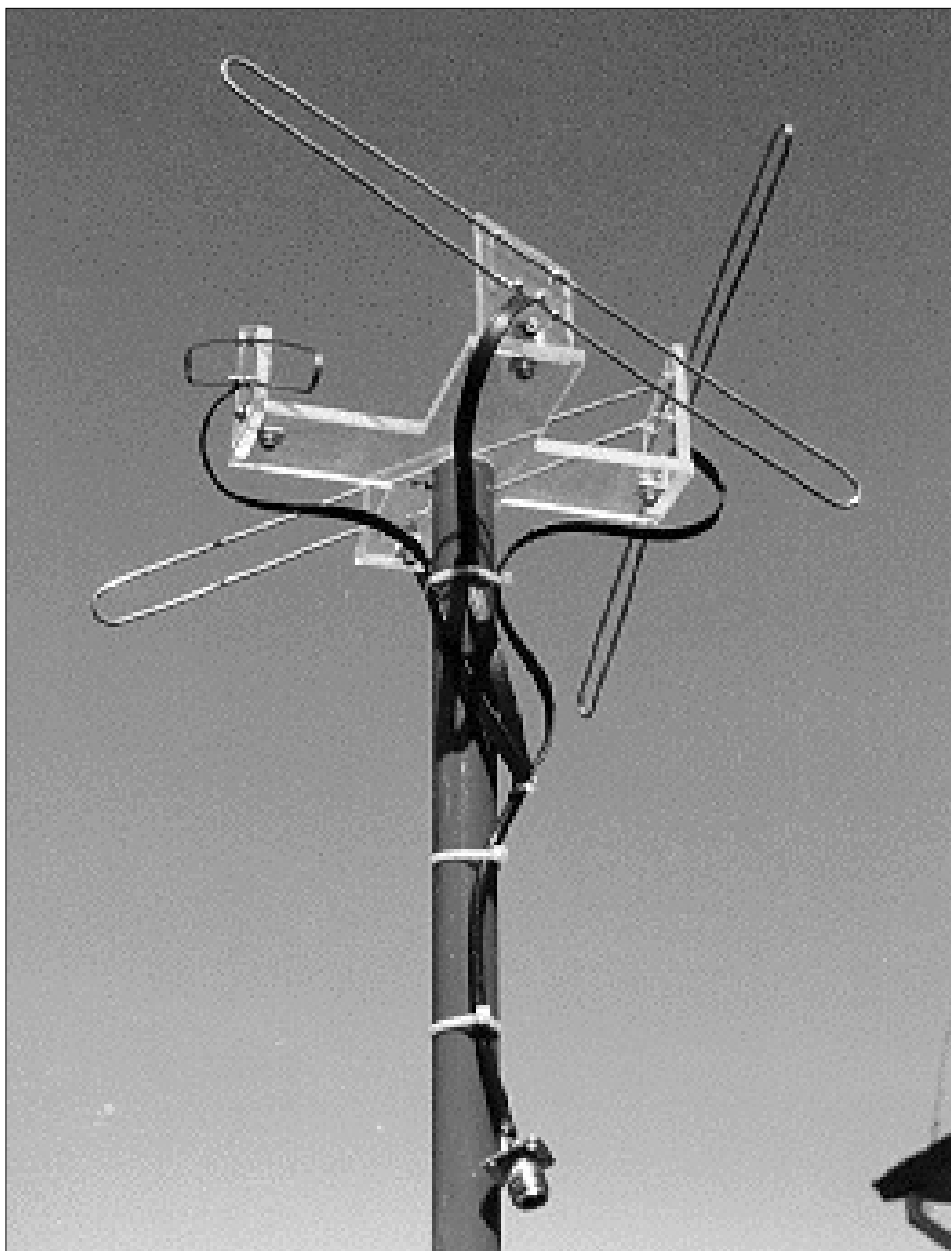


The Quadrafilair Helix antenna is a relatively easy to build, omnidirectional antenna that can be optimized for both Amateur Radio and weather satellite work. (Courtesy: Bob Cash, N8IMO)

In the real world, however, the difference between horizontal and vertical polarization is around 30 to 40 dB. But that's *still* over a thousand times more loss than the difference between RHCP and LHCP!

Also, the number of elements on a Yagi is proportional to its gain. More elements means more gain. However, as in most other things in life, there's a tradeoff between gain and beam width. That is, the higher the gain, the narrower the beam width. So, while a 40-element Yagi may provide excellent gain, it becomes quite another matter to keep it continually pointed directly at a satellite that's rapidly moving across the sky.

The bottom line here is that, while circularly polarized Yagi beam antennas are absolutely wonderful for full coverage satellite work (and I've used my share of them over the years) they are absolutely *not* essential. I've still achieved consistently good results, particularly on the LEO birds, using any number of simple, linearly polarized (hand-held or mounted) Yagi beams and/or eggbeater designs.



The Lindenblad antenna is yet another, relatively easy to build omnidirectional antenna suitable for satellite work. (Courtesy: AMSAT)

My success with linear polarized Yagi antennas is probably because most of our satellites use circularly polarized antennas for their downlinks and they also rotate and tumble through space. So, the practical effects of cross polarization are at least partially minimized by these two factors. This also means that the amount of time when your and the satellite's antennas will be *truly* cross polarized will usually be so brief that the momentary drop in signal strength will most often be imperceptible to your ear.

You can find any number of cheap and easy Yagi antenna designs and construction details on the Internet. A collection of three such articles by Richard Crow, N2SPI, ran

in *The AMSAT Journal* in 2006 and have since been re-published on the AMSAT Web site at: <http://www.amsat.org/amsat-new/information/faqs/crow/index.php>.

Dish Antennas

Dish antennas are the next step up from Yagi antennas. However, the benefits of a dish are often not worth the cost at the VHF and UHF frequencies used by our current amateur satellite fleet. A dish starts to become feasible in the 1.2 GHz range. However, here again, the high gains achieved by a dish antenna are usually offset by the fact that the satellite is rapidly moving across the sky. What's more, complexity (and costs) increase dramatically when using dishes because there are not that many Amateur Radio dish suppliers around.

Usually, you'll have to adapt dish feeds and reflectors manufactured for other purposes.

Feed Lines for Satellite Work

Most veteran satellite operators know that otherwise excellent antennas can be rendered quite useless if they are linked to your station equipment with poor quality feedline. The feedline is what connects your antenna to your radio. And, while the proverbial "wet noodle" feedline might work well for local VHF/ UHF repeater or scanner activity, again, because satellite work is weak signal work, many types of feedlines used in the former activity are *not at all* suited for the latter.

The principal concern with feedlines is *loss*, and every feed line has it to some degree. That is, if you insert 50 Watts into a feed line at your station, you'll have *less* than 50 Watts once your signal gets to your antenna. The rest of the power is lost somewhere in the feedline, usually in the form of heat. Unfortunately, these characteristics are also at work when you receive signals as well. And because the signal from one of our satellites is *already* weak when it strikes your antenna, it follows that you can ill afford to waste *any* of that RF heating up your feedline! What's more, those losses usually increase as the line length and operating frequencies being transmitted or received increase.

So, most of us working the birds these days are using some form of coaxial cable (coax) for feedline. There are about as many varieties of coax as there are companies manufacturing them. However, most of us use some form of low loss coaxial cable such as Belden 9913 or Times LMR 400 as opposed to lengths of RG-58, RG-8X. RG-213 or RG-8 used in most other Amateur Radio work.

Veteran satellite operators use low loss coax cable in their Earth stations because, as the name implies, this coax exhibits much lower losses (particularly at VHF and UHF frequencies) than those used for other (primarily HF) amateur activities. For many years, I've used a variety of Belden 9913 coax for my various satellite stations. It exhibits a relatively low loss (on the order of about 2.6 dB at 400 MHz per 100 feet), which is roughly half that of a similar length of RG-8 coax (about 4.1 dB).

The magic number to always keep in mind when comparing feedlines is 3 dB. That's because, for every 3 dB of loss, roughly *half*



Satellite antennas don't need to be fancy to be effective. Here, a pair of vertically polarized, home brew Yagi antennas made from bits of wire and wood are mounted on a wooden cross boom. (Courtesy: AMSAT)

of your signal is being wasted in the feedline. So, in the example above, at frequencies close to our uplink and downlink frequencies (400 MHz), using a 100-foot length of RG-8 means that *well* more than half of your uplink power (or downlink signal) will be lost in the coax.

Unfortunately, for all of its low loss attributes, Belden 9913 also has a dark side. Because the dielectric in this coax is largely made up of air, it tends to attract moisture. And even though you can try your best to completely seal connection points from the elements, over time, the normal heating and cooling of the atmosphere *will* result in moisture getting into the cable. For this reason, veteran satellite operators sometimes (derisively) refer to 9913 and its variants as "garden hose".

Fortunately, a newer brand of cable on the market, called Times LMR 400, offers about the same loss characteristics as Belden 9913 at about the same price, but without the garden hose issue. An excellent discussion on various types of transmission lines (including their loss characteristics) can be found at: <http://www.hamuniverse.com/coaxdata.html>.

Looking Ahead

In future articles, I'll continue our discussion of innovative ways to optimize your satellite base station, including how to select the

proper connectors for your antenna feedlines and what to look for when selecting a base station radio. I'll also pass along some more tips on how to find and track our Amateur Radio satellites. See you then! 🌐



Introducing AMSAT Fox by Tony Monteiro, AA2TX, aa2tx@amsat.org

Background

The AMSAT Echo satellite, also known as AO-51, has been a huge success for AMSAT. It is the most popular ham satellite in orbit today and with its multiple bands and modes, it offers many interesting communications possibilities. But the basic mode V/U, FM repeater sees the most activity because it can be worked using only simple equipment such as an HT and a hand-held antenna. This capability is a very important entry point to ham satellites and serves as a low-cost way to try out satellite operation before making a big investment in equipment. For this reason, it is often referred to as the *EasySat* mode. It is also clear that even highly experienced satellite operators make frequent use of this mode for demonstrations and in collecting contacts for the VUCC-Satellite and WAS-Satellite awards from the ARRL.

While it is still working reasonably well, AO-51 is over 6 years old and is showing its age. At the recent 2010 Space Symposium, Mark Hammond, N8MH, who runs one of the command stations, presented a paper showing a major loss of battery capacity on the satellite. [1] This is a significant concern and AMSAT needs to put a succession plan in place to assure the continuing availability of the *EasySat* mode.

Unfortunately, launch costs keep going up and up. AMSAT has been very successful in demonstrating that small satellites can be exceedingly useful and now commercial interests have jumped on this bandwagon driving up prices. The days of getting a free or cheap ride on spare launcher capacity seem to be over. When AO-51 was launched, *microsat* technology provided the low-cost path to orbit. This is no longer true. AMSAT could not afford to launch a *microsat* like AO-51 any more even if someone gave us the satellite for free. Given that launching satellites is our life blood, what can AMSAT do?

First of all, if we manage to stay relevant in the satellite world and stay vigilant, there is always the possibility of future good fortune. A great, cheap ride like those of the Phase 3 satellites would fall into this category. Should such an opportunity arise, we would certainly want to be in a position to take advantage of it. But, there is nothing remotely like this on the current horizon and we cannot depend on just waiting for

good luck.

Another possibility is to take advantage of a *rideshare*. Many AMSAT members recall fondly the days of RS-10/11 and RS-12/13. These provided outstanding service when they were operational and yet they were not really *ham* satellites at all; they were hitchhiking communications payloads on Russian navigation satellites. AMSAT has continued to explore new *rideshare* opportunities but so far without success.

That leaves us with *reality*. If we want to launch satellites, we must take advantage of the low-cost paths to orbit that actually exist today. We are extremely fortunate to have access to ISS launches through ARISS. The ARISSat-1 satellite should be deployed in February 2011 from the ISS and is currently the subject of a set of articles in these *AMSAT Journal* pages. We have built several of these satellites and hope to be able to launch a series of them from the ISS. But, while ARISS launches are at no cost to AMSAT, the requirements for manned spaceflight are far more complex than a normal satellite launch. This makes the satellite development more difficult, taking more time and costing us more to complete. And with the ISS in a very low orbit, satellites launched from the ISS have a relatively limited communications range and a short lifetime in orbit. If we want to launch a satellite into an orbit with a longer lifetime and wider communications range, we need to take advantage of CubeSat technology.

CubeSats

A CubeSat is a small satellite standard developed by Professor Bob Twiggs, KE6QMD, while at Stanford and Professor Jordi Puig-Suari from Cal-Poly. Many of you have met Bob as he is especially well known in the ham satellite community and has frequently attended and presented at AMSAT Space Symposiums including the recent one in Chicago. The intent of their CubeSat standard was to allow universities to be able to afford to launch a small satellite for research and education purposes. The key to the low cost is a standard deployment scheme that they called a P-POD (Poly PicoSatellite Orbital Deployer.) The P-PODs allow a launch integrator to easily and inexpensively combine many small satellites onto a single launch so that the average cost per satellite can be dramatically reduced. A

CubeSat launch to LEO can generally be obtained for less than \$100k.

While the CubeSat project may have started small, it has grown tremendously over the past few years with over 40 university satellite launches to date and a rapidly expanding technical community. Even large satellite maker Boeing has built and launched a CubeSat. Besides the low-cost aspect, a key advantage of using CubeSat technology is the availability of regularly scheduled launches. Unlike the *design-to-launch* approach used on previous AMSAT satellite projects, a CubeSat can be designed and built to the standard alone without requiring a predetermined launch.

Further, NASA has now adopted the CubeSat standard and is offering subsidized launches to participating universities. While AMSAT is not an accredited educational institution, education is part of our mission and these subsidized launches offer the potential to partner with a university to fly a joint science and ham radio mission at a further reduced cost. With all the activity in this area, it is clear that AMSAT needs to participate in the CubeSat community if it is to remain relevant in the emerging small satellite world.

While we are a little late to the CubeSat party, there is a big technical hole in this community that AMSAT has the potential to fill. Radio communications, one of our key strengths, is one of the key weaknesses among most university satellite projects. [2] There is very little RF design knowledge or even interest among most university satellite developers. The majority of universities are primarily interested in flying science or technology experiments and the radio communications links are often considered a necessary evil. By demonstrating a better way, we have the potential to vastly improve the reliability and effectiveness of these CubeSat RF links while promoting ham radio and seeding the LEO space with ham communications transponders.

Challenges

Of course all opportunities come with challenges and CubeSats are no exception. One really big challenge is the very small size. A basic (1U) CubeSat is essentially a 10 cm cube or roughly 4" x 4" x 4". This is only about 1/16 the volume of AO-51, which was called a *microsat*! Then there is the weight limit of a little less than 3 pounds;



again a fraction of even the diminutive AO-51. The small size severely constrains the space available for solar panels, antennas and batteries on the satellite. To meet these challenges while still providing excellent performance, cutting-edge technology is needed. This includes the use of low-power, miniaturized electronics, state-of-the-art batteries and solar cells, a high-efficiency power supply with maximum power-point tracking, deployable antennas, deployable solar panels to increase the power available and advanced communications coding and modulation techniques.

The really good news is that these challenges have driven advances in small satellite technology across many fronts. Some of the current research projects in this area include stacking multiple cubes to make a bigger satellite, inflatable antennas, active attitude control including 3-axis stabilization, electric micro-propulsion motors for in-orbit maneuvering and even launches to MEO and GTO. CubeSats are certainly not a dead-end!

In many ways, developing a CubeSat is like building a high-performance sports car. The need to maximize performance under tight constraints can be both exhilarating and highly taxing on an organization's capabilities. These challenges have resulted

in a CubeSat mission success rate of less than 50% not including launch vehicle failures. [3] While there is only limited information on these failures, in many cases it appears to be due to technical errors embedded in the design as was described in the AAU project report. [4]

While AMSAT has historically had excellent technical depth, our engineering process and project management have not been so strong. This is exacerbated by the fact that our satellite developers are all volunteers and widely dispersed covering many states and time zones. With little margin for error in a high-performance project like this, we will need to make improvements in these areas if we want to be successful.

The AMSAT Fox Project

The Fox project was originally recommended by the Engineering Task Force at the 2009 AMSAT Board of Directors meeting. The board approved the project but unfortunately the domino effect of delays in the ARISSat-1 program caused the start of Fox to be delayed as well. Now that ARISSat-1 has been shipped, Fox satellite development has begun.

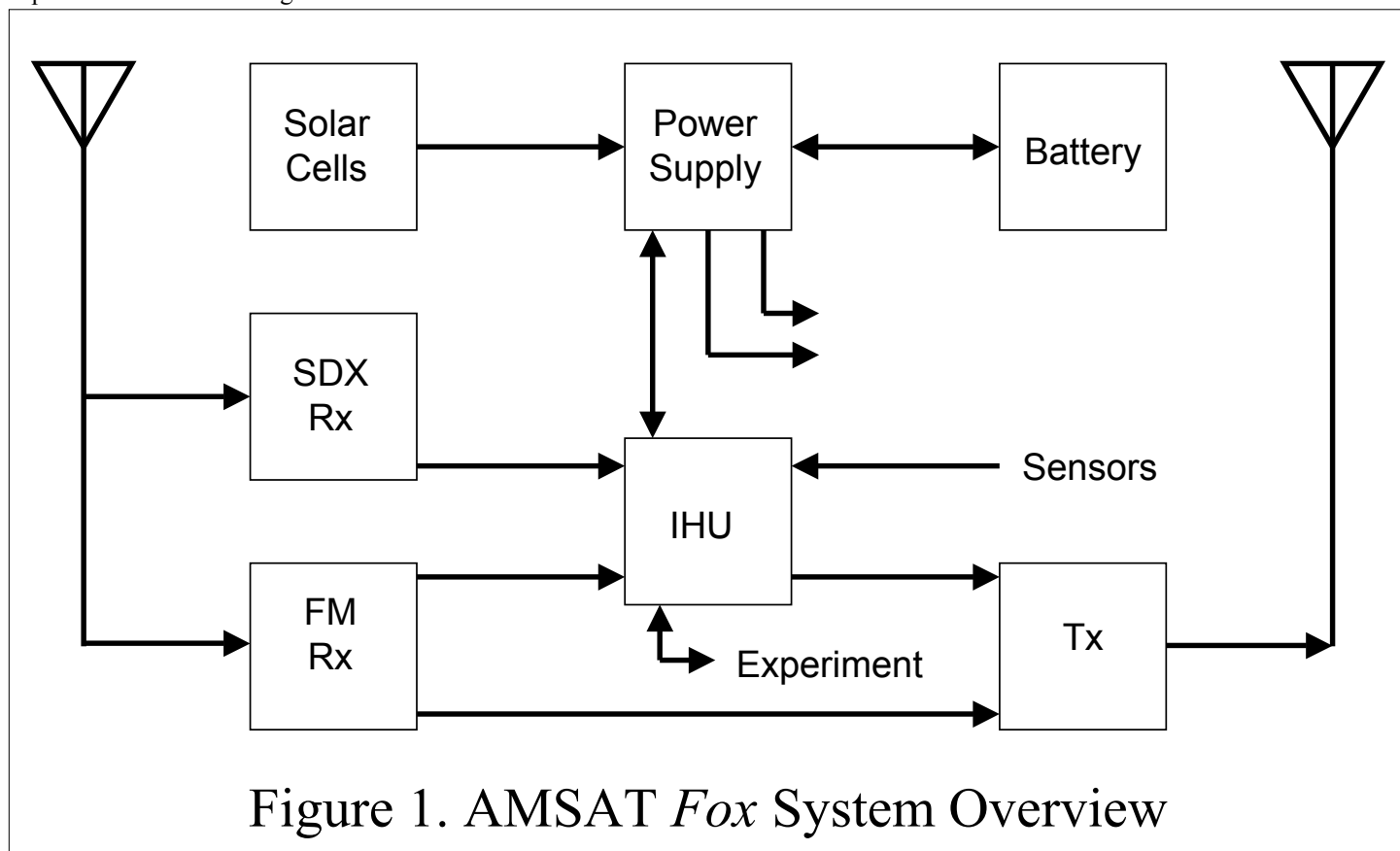
The primary mission of the Fox satellite is to provide reliable communications using simple ground station equipment such as

an HT and a hand-held antenna. This is essentially a replacement for the AO-51 EasySat FM repeater mode.

An important secondary objective is to develop the Fox design as a general purpose satellite bus that can be reused to fly university science missions and future AMSAT missions. A key component is its software defined transponder (SDX.) The SDX will support advanced modulation and coding techniques, providing for high reliability of the radio links. For the initial Fox mission, this will be used primarily for telemetry and satellite ground control. But, since the SDX functions are defined in software, it can be programmed to provide a linear or digital transponder on a future mission using the same hardware design.

By demonstrating a superior RF communications capability, we hope to be able to make a significant contribution to the CubeSat technical community in an area that has traditionally been a weakness. This is also helpful in facilitating AMSAT's education outreach efforts and in promoting closer working relationships with universities.

In this same vein, AMSAT has agreed to mentor a group of 34 senior engineering students at the State University of New York (SUNY) in Binghamton. These students



are working on aspects of the Fox satellite design as their senior design projects. Alex Harvilchuck, N3NP, is leading this effort with Bob Davis, KF4KSS, Dick Jansson, KD1K, Stan Wood, WA4NFY, Lou McFadin, W5DID, Bill Reed, NX5R, Joe Fitzgerald, KM1P and Tony Monteiro, AA2TX, all volunteering as student mentors.

Status

The first Fox project meeting was held in October at the 2010 AMSAT Space Symposium to discuss technical ideas and concepts for the new satellite. In attendance were Bob McGwier, N4HY, Bill Reed, NX5R, Drew Glasbrenner, KO4MA, Bryan Klofas, KF6ZEO, Barry Baines, WD4ASW, Bob Davis, KF4KSS, Dan Schultz, N8FGV, Alex Harvilchuck, N3NP, Phil Karn, KA9Q, Lou McFadin, W5DID, and Tony Monteiro, AA2TX.

These ideas were documented by Alex and Bryan and were used to produce a Satellite Spec Sheet that formed the basis for getting the SUNY students started. The project team has also used this to start work on a formal System Requirements specification. As of this article, a draft specification has been produced and is currently being reviewed by the team. This will be the first AMSAT satellite to have a documented set of requirements and is a first step in improving our engineering process.

Some Fox highlights are as follows:

- 800 km target orbit
- Mode U/V (mode B)
- FM Transponder
- Software Defined Transponder (SDX)
- Digital up/downlinks via the SDX for telemetry and ground control.
- 500 mW transmitter
- Deployable Solar Panels
- Passive Magnetic Stabilization (like AO-51)
- Space and power for a science experiment (i.e. "this space for rent")

A high-level system overview of the satellite can be seen in Figure 1.

AMSAT Needs You!

AMSAT depends on volunteers to develop our spacecraft. As of this time, we have had a good response to this project in signing up both experienced old hands as well as adding some new names to the satellite project team. The current team is shown in Figure 2. This is a great start, but we still need volunteers to work in a variety of areas. Come and join us! Some of the open areas are as follows:

- Radiation Engineering
- Systems Engineering
- Power System Design
- Printed Circuit Board Design
- Satellite Construction
- System Test
- Test Equipment Development
- Project Management
- Ground Station Software
- and more!

If you are interested in working on our new Fox satellite program, please send an email to me at aa2tx@amsat.org. Note that due to the US State Department's International Traffic in Arms Regulations (ITAR), you must be a US citizen or US permanent resident to work on our satellite development projects.

Summary

Fox is AMSAT's latest satellite project and our first CubeSat. Fox will provide a variety of capabilities including an FM transponder, a software defined transponder (SDX) and the ability to host science and technology experiments. While the initial mission is primarily to provide a successor to AO-51 and the EasySat FM transponder mode, the satellite design is intended to be re-usable and could be used to fly a linear or digital transponder in a future mission.

As part of our educational outreach, AMSAT will be mentoring engineering students from SUNY-Binghamton who will be contributing aspects of the Fox satellite design as their senior design projects. We expect to show these student contributions at the Dayton Hamvention in 2011.

References

- [1] *AO-51 Battery Capacity and Spin Trends*, by Mark Hammond, N8MH, and Colin Hurts, VK5HI; Proceedings of the 2010 AMSAT-North America 28th Space Symposium, Pgs 122-134.
- [2] *A Survey of CubeSat Communication Systems*, by Klofas, Anderson and Leveque; presented at the CubeSat Developers Conference - April 2008
- [3] Clyde Space - Past Missions; <http://www.clyde-space.com/resources/cubesats>
- [4] *Educational Value and Lessons Learned from the AAU-CUBESAT Project* by Alminde, Bisgaard, Vinther, Viscor and Stergard; Department of Control Engineering; Aalborg University; Denmark. <http://vbn.aau.dk/files/168186/fulltext>

Team Member	Call	Primary Area
Keith Baker	KB1SF	PR & Project Mgt.
Steve Bible	N7HPR	RF Card Design
Larry Brown	W7LB	Satellite Construction
Bob Davis	KF4KSS	Mechanical Design
Joe Fitzgerald	KM1P	Software
Bdale Garbee	KB0G	IHU Card Design
Alex Harvilchuck	N3NP	SUNY Program Leader
Dick Jansson	KD1K	Thermal Design
Jim Johns	KA0IQT	Software
Phil Karn	KA9Q	Communications Software
John Klingelhoefter	WB4LNM	RF Card Design
Bryan Klofas	KF6ZEO	Systems Engineering
Steve Marrano	KC2VNI	System Test
Lou McFadin	W5DID	SUNY Mentor
Bob McGwier	N4HY	Technical Advisor
Tony Monteiro	AA2TX	Project Mgt.
Robinson Nunez	none	Thermal Design
Keith Packard	KD7SQG	IHU Card Design
Mike Radwick	N8XKZ	Software
Bill Reed	NX5R	Software
Jay Schwartz	WB8SBI	Software
Stan Wood	WA4NFY	Antennas

Figure 2: AMSAT Fox Project Team



More on Converting California Amplifier 31732 Downconverters for 2.4 GHz

by John M. Franke, WA4WDL, jmfranke@cox.net

Much has been written concerning how to convert California Amplifier model 31732 MMDS (Multichannel Multipoint Distribution Service) downconverters to receive Amateur Radio satellite signals on 2.4 GHz. This article is an adaptation of one method. But first, it is important to understand why the model 31732 is a good choice among a seemingly endless variety of MMDS downconverters.

A large number of the MMDS downconverters were designed for the 31 MMDS channels between 2.5 GHz and 2.686 GHz. These downconverters, including the models 130001 and 130002, use one industry standard local oscillator frequency of 2278 MHz, which results in an IF frequency range of 222 MHz to 408 MHz. Unfortunately, the IF range does not include the 144 or 432 MHz bands which would allow amateurs with rigs for those bands to conveniently use them as IFs. Changing the downconverter crystal to yield a 144 MHz band IF can be done, but the IF response is very poor at 144 MHz. The downconverter IF filter pass band can be shifted, but the work is not an easy undertaking.

The model 31732 downconverter was designed for 33 channels, the previously mentioned 31 plus two MDS channels between 2.15 GHz and 2.162 GHz. To reduce costs the model 31732 downconverter uses the same 2278 MHz local oscillator for both bands. The second RF band results in a second IF frequency of 116 to 128 MHz. However, in practice, the net result is one IF band extending from 55 MHz to 408 MHz, which covers the entire amateur 144 MHz band. Two things must be done to use the downconverter on 2.4 GHz. First, the internal crystal must be replaced. Second, the RF preamplifier band pass filter must be retuned or shifted to cover 2.4 GHz.

Changing the internal crystal is very easy and can be done without removing the printed circuit board from the case. The downconverter has a phased locked loop to lock the local oscillator VCO to 2278 MHz, the 256th harmonic of an 8.89844 MHz crystal. Replacing the crystal with an 8.8125 MHz crystal will result in a local oscillator frequency of 2256 MHz and 2.4 GHz being converted to 144 MHz. Tests with the local oscillator PLL indicate no tuning

or other circuit modifications are required for crystals between 8.465 and 9.148 MHz. Information for specifying/ordering a crystal can be found at: <http://www.qsl.net/n9zia/wc0y/index.html>.

Retuning the RF preamplifier bandpass filter is more difficult, but can be done by most amateurs. There are two approaches. The first involves positioning and soldering small tabs to the individual filter element traces and is well illustrated by Adam Moss, G0ORY, at: <http://www.qsl.net/n9zia/31732/index.html>. However, this method requires a bit of skill with delicate soldering and lots of patience. The second method involves dielectric loading of the filter to shift the pass band to a lower frequency and is illustrated by Ward, WC0Y, at: <http://www.qsl.net/n9zia/wc0y/index.html>. The method involves overlaying the bandpass filter with strips of adhesive backed Teflon tape. Layers of the tape are added until the desired response is obtained. I had a hard time finding a supply of the tape, so I devised an alternate method using thin uncoated Teflon sheets and a clamp made from a piece of bare 1/16" inch thick G-10 glass-epoxy printed circuit board. My adaptation is easily done and is reversible or easily changed for other frequencies. An added plus was the fact that I had all of the parts in my junk box.

Even in the sticky back Teflon tape conversion, the lower frequency MDS band RF filter is disconnected by a well placed cut to the circuit board. Otherwise the overall noise figure would be adversely affected by noise and signals passing through the low band RF channel. I prefer removing the low

band RF filter completely.

Two solder blobs on the MMDS RF bandpass filter strip lines were carefully shaved down by hand using a single edge razor blade so the Teflon sheets would lay flat against the filter traces. As with the sticky tape conversion, I built up the thickness of Teflon by adding layers until the desired response was obtained. It was fun seeing a signal from my generator improve by a little over 30 dB as the layers of Teflon were added and pressed down in contact with the filter elements. My best results were obtained with a stack of four 0.010" thick sheets. The individual sheets measure 1" by 1.5". The layers are clamped in place with the fiberglass plate. The plate was fabricated from a completely etched piece of G-10 printed circuit board stock. A hand nibbling tool was used to make notches and cutouts where the plate passes over the local oscillator bandpass filter, the mixer striplines and surface mounted components. The plate is held in position with screws through original circuit board mounting holes. Two metal 4-40 washers are placed between the glass-epoxy cover plate and the circuit board at each screw location to act as spacers and reduce warping stresses. Together, the two washers closely match the thickness of the stack of Teflon sheets. I inserted headless 4-40 screws into each mounting hole to hold the washers in place while the cover plate was gently dropped in place over the stack of Teflon sheets. The headless screws act

continued on page 13 ...

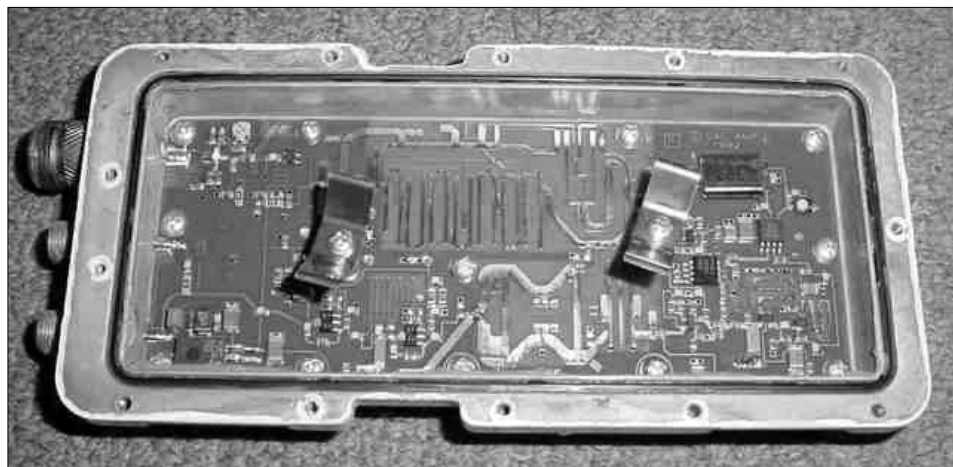


Photo 1: Downconverter with cover removed. The low band, MDS filter is at the top edge and the multiple hair-pin high band, MMDS filter below the low band filter.

Bill Ress, N6GHZ, Appointed AMSAT Director of Launch Opportunities

AMSAT President Barry Baines, WD4ASW, has appointed Bill Ress, N6GHZ, of Camino, CA as AMSAT's Director of Launch Opportunities, a new position. The appointment was announced at the AMSAT BoD meeting and AMSAT Annual Meeting that was held in October, 2010 in Elk Grove Village, IL. In his capacity as Director of Launch Opportunities, Bill will serve as the central point for managing:

- Fox Launch Opportunities
- Ride Share Opportunities
- AMSAT Presence at small satellite conferences

Supporting Bill's efforts will be a volunteer task force to help coordinate and compile launch information. Working with the VP-Engineering and AMSAT President, the Director will make recommendations to the AMSAT BoD to develop launch objectives and serve as the central point for securing feasible launches for AMSAT spacecraft.

AMSAT President Barry Baines, WD4ASW, noted Bill's qualifications for this position. "As a former BoD member who chaired the Engineering Task Force that developed the recommendation to the AMSAT Board of Directors to build a CubeSat ('Project Fox'), Bill understands AMSAT's strategic vision, our current capabilities and the current state of flight options. His professional contacts and location on the West Coast places him in an excellent position to identify and nurture potential launches, a significant task that requires an appreciation of both AMSAT and the state of the launch industry."

Bill and his team will be working to establish dialog with both satellite customers who may have ride share potential as well as satellite builders to evaluate potential ride share potential. The team will also evaluate and determine how best for AMSAT to respond to NASA Launch Initiatives, other launch provider announcements, and potential university relationships for securing launches as these opportunities arise. Individuals who become aware of potential launch opportunities which might be of interest to AMSAT are encouraged to contact Bill (n6ghz@amsat.org).

Support AMSAT

New Version!

AMSAT is the North American distributor of SatPC32, a tracking program designed for ham satellite applications. For Windows 95, 98, NT, ME, 2000, XP, Vista, Windows 7.

Version 12.8a is compatible with Windows 7 and features enhanced support for tuning multiple radios.

Version 12.8a features:

- Select 2D/3D map with a single mouse click
- Select Political/Physical map with a single mouse click
- AOS voice announcement
- Countdown program displays next AOS of your favorite satellites
- Automatic Internet download of Keplerian elements.
- Graphical tracking on a world map with two selectable Zoom factors.
- One-click satellite switching using 12 "letter" buttons.
- Automatic rotor control support for: LVBTracker, Kansas City Tracker/Tuner, FODtrack, Uni_Trac, SatEL, Labjack U12/U3/Piggyback, W0LMD Mini/Junior/Senior Trackers, Yaesu GS-232, ARS from EA4TX, EGIS rotors, IF-100, RIF-PC, AMSAT-DL interface, WinRotor32, HalloRotor, and WiSP DDE clients.
- Doppler tuning with on-screen frequency displays. Radio models supported: Yaesu FT-847, FT-736R, FT-817, FT-857, FT-897 ICOM IC-820H, IC-821H, IC-910H and other ICOM radios Kenwood TS-790 and TS-2000 DDE client programs
- "Transparent" VFO-knob tuning in addition to keyboard and mouse tuning.
- "Virtual" VFO tracking for half-duplex transceivers FT-817, FT-857, FT-897, IC-706 and IC-7000.
- Includes a special version for ISS split-frequency operation.
- Controls subaudible tone on FT-847, FT-736R, FT-817/857/897, TS790A and IC-910H.
- Includes a non-graphical version and programs that output lists to the screen, printer, or a file.
- Interfaces with WiSP, with user-defined satellite switching priority.
- Includes a separate program for tracking the Sun and Moon.

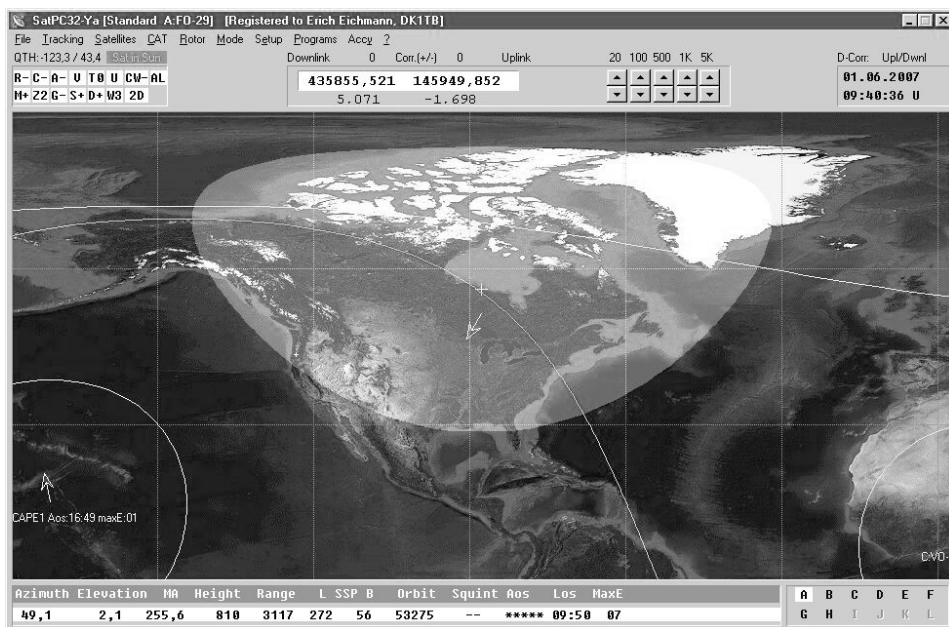
Minimum Donation is \$45 for AMSAT members, \$50 for non-members, on CD-ROM.

A demo version may be downloaded from <http://www.dk1tb.de/indexeng.htm>

A registration password for the demo version may be obtained for a minimum donation of \$40 for members and \$45 for non-members.

Order by calling 1-888-322-6728.

The author DK1TB donated SatPC32 to AMSAT. All proceeds support AMSAT.



as alignment pins. With the clamp plate in place, the headless screws are removed one at a time and replaced with normal 3/8" 4-40 screws. The original screws are now too short. There is a split ring lock washer and a flat washer under each of the three clamp plate screw heads.

Photo 1 shows the model 31732 downconverter with the cover removed. The MMDS and MDS RF preamplifier filters are shown at the top of the image. The low band, MDS, filter is at the top edge and the hair-pin section high band, MMDS filter is below the MDS filter. Photo 2 shows the downconverter after I have removed the low band RF filter by cutting and peeling away the traces using a single edge razor blade and a pair of needle-nosed pliers. The two grounding clips and posts have been temporarily removed for clarity. The posts and clips were reinstalled for actual use. The three headless screws for holding the spacer washers and aligning the clamp plate can be seen. Photo 3 shows the modified downconverter with the fiberglass clamp plate holding the sheets of Teflon in place.

I was able to measure the downconverter image response for IFs of 144 MHz and 123 MHz. I plan to use 123 MHz for at least one application. In both cases, the image response is at least 28 dB down from the response at 2401 MHz. The response is flat from below 2400 MHz to above 2500 MHz.

Now, I need to work on my antenna and start listening for AMSAT AO-51's S-band signal. 📡

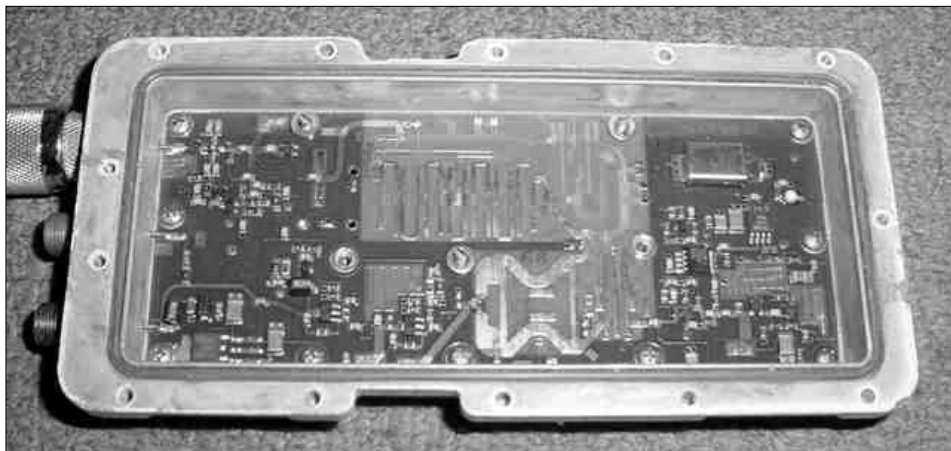


Photo 2: Downconverter with the low band RF filter peeled away. The grounding clips and posts have been temporarily removed for clarity. The three headless screws for holding the spacer washers and aligning the clamp plate can be seen.

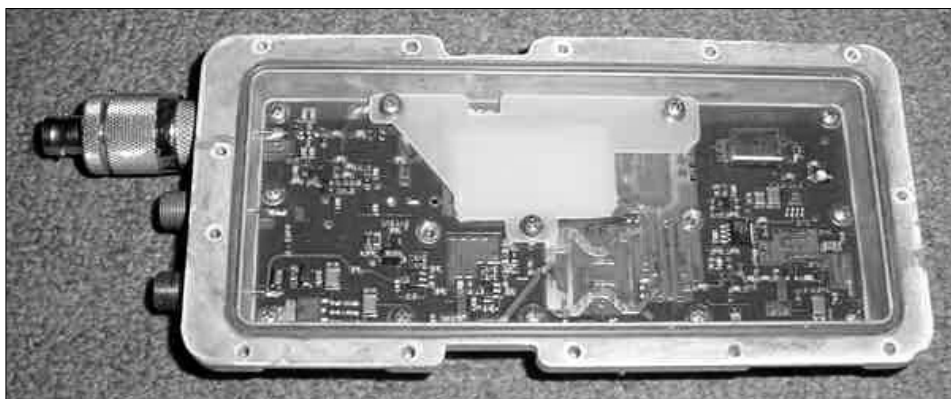
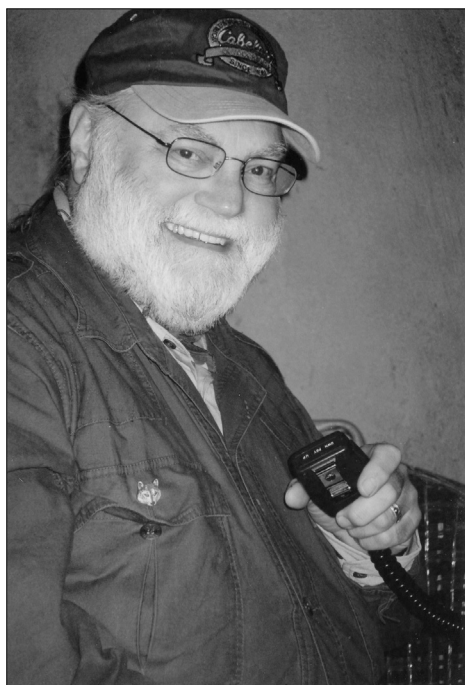


Photo 3: The modified downconverter with fiberglass clamp plate holding the sheets of Teflon in place.



John Champa, K8OCL, SK

On Friday, November 12th, 2010 John Champa, K8OCL, became a Silent Key. He was living in Richardson, Texas and passed away as a result of complications of cancer.

John was a life member of AMSAT and was AMSAT Executive Vice President from 1983 to 1991. He hosted the 10th Annual Symposium in Detroit, MI, his home at the time. John was an officer in the U.S. Army obtaining the rank of Captain and it was my pleasure to be with him at Ft. Gordon, GA where he was the president of the Fort Amateur Radio club and I was his assistant as the VP.

His recent activities included PropNET and high speed telecommunications. He was a member of NARTE, ARRL, VFW and member of many conservancy organizations.

Of his four children, only his daughter Rebecca McLaughlin, KE5LVN, and his son Patrick Champa, KC8LQA, became amateur operators. He will be missed by all and I can still hear him giving me my "first day" contact on OSCAR 13.

Dee Interdonato, NB2F 📡



El Cheapo RX Antenna for AO-7 Mode A Operation

by John Neeley, W6ZKH, w6zkh@att.net

For the past year and a half I have made attempts to work the “Grand Old Lady, AO-7” on Mode A with the downlink frequency of 29.450 Mhz. The uplink was no problem, but trying to come up with a decent downlink antenna has been a challenge at best. I started out with the proverbial G5RV dipole, which proved worthless. Then I went with a standard wire dipole cut to the frequency, to no avail. Next I tried a full-wave loop lying on the roof of the house. That too was not much better than any of the others. My last attempt was building a vertical dipole, which worked better than any of the others. But again, I wasn’t satisfied and knew a simple solution had to be available.

Then a couple of weeks ago I came across an article on QSL.net dealing with a commercial made Moxon antenna and that got my curiosity up. I went onto the Google search engine and found a multitude of articles dealing with the Moxon, but the one Web site I found that dealt with more information regarding this antenna was KD6WD’s Moxon Antenna Project site: www.moxonantennaproject.com.

Bringing up this Web site you arrive at the home page and below the BIG name header is a series of “click-on” sections many of which all have great information, but the ones that are most relevant are the Design and Construction areas. Clicking on the Design button will give you good information and also a calculator providing the measurements required for whatever frequency and type of materials you will be using, i.e., wire, tubing, etc. Download the MoxGen.zip file which will give you the calculator.

Clicking the Construction button will give you various methods other hams have used to make this antenna – from the hubs to the spreaders.

I am not going to try and re-invent the wheel here but give you an idea of what I did to make a temporary model to try out on Mode A. There is a 10 meter Moxon made by M0MRR, which basically I copied with some modifications. M0MRR used hardware store white PVC water pipe and #12 insulated wire. Basically I built the same design, but added some extra PVC on the ends to make it stronger, even though it did increase the weight by a pound or so. This made the ends droop a little more than I wanted, so I ran some heavy fishing line through the mount of the Yaesu elevation rotor and used it as a pivot thereby raising each end up. Yes, I know, it is rather crude, but this project was not meant to be permanent – well sorta.

In my version of this antenna I used a 5 foot long, 1 inch inside diameter schedule 40 PVC pipe for the boom. To each end I glued a 1 inch “T” fitting and then installed two, 1” to ½” reducers. Each ½” PVC element is glued into place at the reducer. Taking dimensions from the calculator mentioned above, gives you the length of each element. I then mounted an SO-239 coax connector onto a piece of aluminum and screwed it to one end of the boom. Noting that the drawing associated with the calculator gives you the length of each leg for the driven element and also the reflector. To attach the #12 wire to the PVC, I used zip ties, but you can also use several wraps of electrical tape instead of the ties.

Once I had the antenna built on the ground,

I then mounted my version about 18 inches below the elevation rotor (see photo), which is only 8-9 ft above ground. Since I am running an end-mounted, single Arrow dual-band Yagi for V/U satellite work, I didn’t have to worry about that antenna hitting the Moxon. From the Moxon to my Yaesu FT-847 receiver I am using 50 ft of RG-8x coax and no pre-amp at the antenna.

Now, for results – I was truly amazed at the performance of this antenna, even at a low elevation surrounded by homes. I can now hear signals I’ve never would have thought of before with the other various antennas I have tried. The east coast passes are received at fairly decent signal levels.

Not having any antenna analyzers for testing of directivity and front to back readings, I used a 10 meter FM repeater output on 29.6 MHz from the Los Angeles area about 150 miles away. The antenna seems to be quite directive, but for the F/B test, I could hear it almost as well off the back as the front. This is probably not the design and may be my construction methods, but this is unknown. There is a claim of 5.1db gain with 25db F/B, but again I have no test equipment to verify this. All I can say is that this antenna works, and that is what I was striving for when building it. I have thought of improving the construction by using some sort of tubing, but at this time I am content with just using it as it is.

So, if you are interested in AO-7 and you have been working Mode B, build this antenna and give Mode A a try. I estimate that I have approximately \$10 in this antenna. That is why I called it, the “El Cheapo”. 📶



The finished antenna.



Feedpoint view.



LVB Tracker Box

- Reasonably priced and all profits go to AMSAT
- USB or serial interface
- Ethernet interface capable
- Open software
- Open architecture
- Built-in rotor interface cable with DIN connector
- Power supplied via the rotor controller cable
- Bi-directional interface allows the unit to read the rotor position
- LCD position readout
- Front panel manual control
- Self-programmable, allows for easy software upgrades
- Supported by the major tracking programs
- Uses GS-232A or Easycomm I/II protocols

Contact Martha at the AMSAT office 301-589-6062 to order.

The complete units will be built in groups, so get your orders in early. Units will be shipped as they become available.

Minimum donations requested for various configurations:

- 1) Bare board - \$20 + S&H - no parts kits available
- 2) Populated board, serial interface - \$100 + S&H - no kit with USB unit, rotor cable and LCD available, we do have a parts list with recommended part numbers and sources.
- 3) Custom Ten-Tec Enclosure powder coat, silk-screened, drilled/punched - \$50 + S&H
- 4) Complete unit - board, USB output, rotor cable, LCD, enclosure - \$200 + S&H **our most popular

Documentation for the LVB Tracker can be found at: <http://www.LVBTracker.com>

Boards and complete units may be ordered from the AMSAT office at 301-589-6062 or from martha@amsat.org

THE R.F. CONNECTION

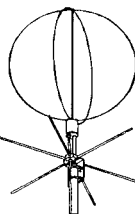
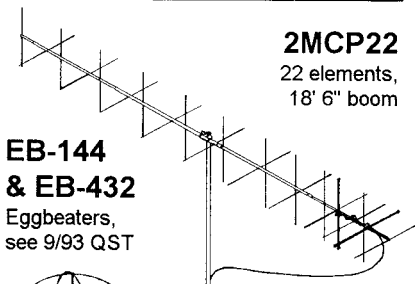
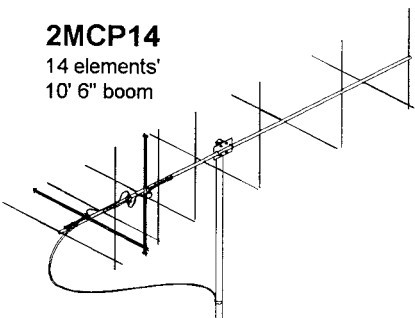
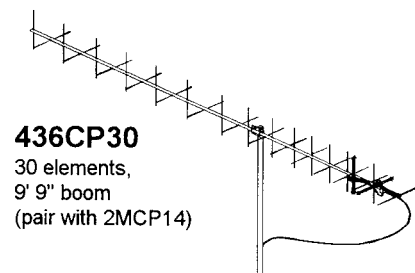
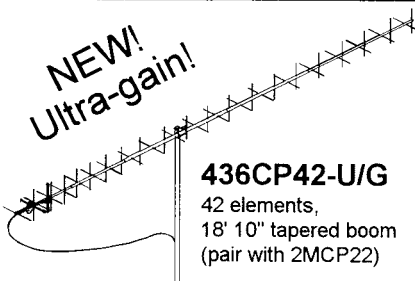
"Specialist in
R.F. Connectors and Coax"
<http://www.therfc.com>

301/840-5477
Fax 301/869-3680
E mail: rfc@therfc.com

Order Line 800/783-2666
Suite 11, 213 N. Frederick Ave.
Gaithersburg, MD 20877

M² YOUR SATELLITE ANTENNA SOURCE

NEW!
Ultra-gain!



- Stacking frames
- AZ & EL positioners
- Fiberglass crossbooms
- Power Dividers
- Phasing harnesses

M² 4402 N. Selland
Fresno, CA 93722
(559) 432-8873 FAX: 432-3059



The BPSK1000 Telemetry Modem for ARISSat-1

Phil Karn, KA9Q, ka9q@amsat.org



Phil Karn's original paper was presented at the 2010 AMSAT Space Symposium held October, 2010 in Elk Grove Village, Illinois. This paper has been updated by the author. The original version is published in the Proceedings of the 2010 AMSAT Space Symposium.

Introduction

The ARISSat-1 (formerly SuitSat-2) spacecraft is planned for launch on a Russian Progress supply flight to the International Space Station (ISS) sometime in 2011 where it will be deployed as a free flying satellite.

ARISSat-1 will carry a new telemetry modulation and coding scheme, BPSK1000, designed to handle the severe fading often encountered with low orbit satellites without attitude control. Its performance and the link budgets for the ARISSat-1 spacecraft are such that reliable reception should require only a simple whip or ground plane antenna, a conventional 2 m SSB receiver and a reasonably modern personal computer with audio A/D input.

BPSK1000 uses differential binary phase shift keying (DBPSK) at a channel symbol rate of 1 kHz in a SSB bandwidth. With constraint length 7, rate $\frac{1}{2}$ forward error correction (FEC), the user data rate is about 500 bits/sec. HDLC framing provides application flexibility (including the ability to carry AX.25 in other applications) and a deep (16 second) convolutional interleaver provides strong protection against fading.

I hope that the demonstrated performance of BPSK1000 on ARISSat-1 will encourage other amateur satellite projects with similar needs to consider it. At the very least, I hope to convince other satellite designers that forward error correction (FEC) should be a standard feature of *every* amateur satellite data link!

Software to demodulate and decode the ARISSat-1 BPSK1000 beacon will be available for Microsoft Windows, Mac OSX and Linux. I wrote the reference BPSK1000 demodulator and decoder in C under Mac OSX and Linux as libraries and command line tools and I am distributing them under the terms of the GNU General Public License (GPL). I strongly encourage everyone to study, modify, enhance, test, break, fix, improve, experiment with and use what I have written.

My reference decoder is incorporated into the Windows program by Doug Quagliana, KA2UPW, and the Mac OSX program by Gilbert Mackall, N3RZN, both of whom have turned a set of command line tools into

complete, user-friendly applications.

My software can use, but does not require, multiple CPUs and the AltiVec (Power PC) or SSE (Intel/AMD) vector instruction sets to improve performance, in some cases rather dramatically. Multiple CPUs are used through the POSIX Threads (pthreads) library, and vector instructions are invoked with gcc C compiler intrinsics; there is no actual assembly code.

The rest of this paper discusses the physical layer modulation, interleaving, coding and framing elements of the generic BPSK1000 air interface. Its specific use by ARISSat-1 is documented separately.

Requirements

Engineering is all about making tradeoffs and the design of a digital modem is no exception. Every engineering project should start with a clear list of requirements. For a modem, making and meeting those requirements requires a good understanding of the channel on which it is to operate.

Since classroom educational demonstrations are a prime goal of ARISSat-1, my goal was to design a signal format that, while as robust and efficient as possible, should not require any special hardware beyond a general purpose personal computer and a 2 m receiver or transceiver. Nor should it require any special knowledge, training or practice beyond ordinary ham radio skills. I especially did not want to assume that the operator was already an experienced amateur satellite operator. I want to make it possible for *any* ham with the necessary equipment, not necessarily one already familiar with amateur satellites, to do a successful classroom demonstration. My hope is that this may entice more kids to become hams, and more hams to join AMSAT.

No one gave me a formal list of requirements beyond "it must work" and "we must have it before such and such date", so I made my own:

Requirement #1: No special RF equipment needed.

It must be possible to receive the ARISSat-1 telemetry with a stock 2 m SSB Amateur Radio transceiver. As nice as it would be to use a software defined radio capable of wideband operation, that would be more

appropriate for a future mission, not one intended primarily for school demonstrations and general education.

Requirement #2: Good power efficiency.

DC power is at an absolute premium on any spacecraft and downlink transmitters invariably take the lion's share of that power. The downlink mode must be as power-efficient as possible.

Additionally, it should be possible to reliably receive ARISSat-1 telemetry with a small and simple antenna such as an ordinary whip or ground plane. Steered directional antennas should not be required.

Although it would be nice to work with FM-only transceivers, FM is just too inefficient for satellite use. Implementing an efficient modulation mode in software requires a linear transceiver and that currently means SSB. This leads to the requirement that the signal fit within the typical SSB voice filter response of 300 to 2700 Hz, a bandwidth of 2400 Hz.

A prime personal goal of this project – as with my previous effort that put an experimental FEC format on AO-40 – is to demonstrate that there exist far more power-efficient modulation and coding methods for amateur satellite telemetry links than are generally flown. Such improvements can either cut the cost of constructing a satellite by reducing the amount of DC power it must generate, and/or it can make amateur satellites accessible to more hams by reducing the size, cost and complexity of the ground antennas. Many hams live in homes with antenna restrictions where the traditional 2 m/70 cm Yagi combination simply isn't an option.

The AFSK/FM mode used by terrestrial amateur packet radio is *particularly* inefficient. Quite frankly I am at a loss to understand why anyone would fly it on a spacecraft.

Because forward error correction (FEC) codes can provide a significant "coding gain" that reduces the transmitted energy per bit, this requirement alone dictates the use of FEC, even without the additional dramatic improvements it provides in fading (see below). It has long been easy to achieve



coding gains of up to 7 dB on channels impaired by only Gaussian (thermal) noise, and the latest codes can provide gains of up to about 10 dB.

Requirement #3: Fade tolerance.

Closely related to the previous requirement is the need to tolerate the unpredictable and often deep fades often encountered with amateur satellites in low earth orbit. Although the line of sight from satellite to ground station may not be interrupted, amateur spacecraft (such as ARISSat-1) are typically uncontrolled or partly controlled in attitude (e.g., with bar magnets) with the result that antenna nulls can sweep across the ground station and produce deep nulls. Multipath fading can also occur when omnidirectional ground antennas pick up reflections of the satellite signal from the ground or nearby buildings.

This requirement is closely related to #2 because an inefficient link (i.e., one without FEC) requires a lot of excess power to ride through a fade. An alternative is to keep retransmitting until the data gets through, but all those unsuccessful transmissions are also wasted energy. Fading represents errors that can be corrected by FEC, but not all FEC codes can tolerate the long bursts of errors associated with slow fading. Therefore, we need either FEC suited for burst error correction or an interleaver to break up error bursts into a long series of single bit errors that can be corrected more easily.

To see the huge benefit of FEC on fading channels, consider that without coding you need enough link margin – excess transmitter power – to ride through the usual fades; i.e., it must be designed for the worst case. Since the channel is faded for only a small fraction of the time, most of the time all that excess power goes to waste. On the other hand, a well designed FEC scheme for a fading channel will operate as long as the *average* signal-to-noise ratio is high enough. Brief fades, even when infinitely deep, can be reconstructed by the decoder from redundant information received when the channel is not faded. This difference between average and worst case SNR becomes the effective coding gain of the FEC. When the fades become very deep the coding gains can become truly dramatic. The link simply

won't work without it.

Requirement #4: The average amateur should be able to tune the signal by ear without specialized skill or an excessive amount of practice.

Doppler shift from a LEO satellite on 2 m is about +/- 3.3 kHz, so tuning of a SSB receiver is obviously required throughout the pass. As desirable as computer controlled tuning may be, I did not want to require it.

I probably spent more time thinking about how best to meet this requirement than all the others combined. The problem is that efficient digital modes tend to sound like band limited white noise, and white noise is notoriously hard to tune to an exact frequency! [1]

Fortunately, an elegant and effective solution presented itself. The ARISSat-1 downlink also includes a CW beacon and I recommended that it simply be moved to the lower null of the digital telemetry signal. Here it will not interfere with the beacon and vice versa, but it can easily be heard and tuned by a human operator in such a way that it automatically tunes the telemetry signal as well.

Requirement #5: the transmitter must be simple.

Although ARISSat-1 carries a lot of processing power by satellite standards, CPU cycles and memory are both still quite limited in comparison to a typical personal computer. The IHU and DSP have many other things to do besides encode telemetry. Even when there's plenty of real time, you still want to minimize the number of executed instructions because every one of them requires some amount of energy to execute.

The only element of signal generation that created any concerns was the BPSK modulation. Not the modulation per se, but the filtering associated with it. The FIR filter is 454 taps long, but not every tap needs to be computed on every sample. By signaling with impulses most of the delay line will contain zeros, allowing those multiplications to be skipped.

Fortunately, generating BPSK1000 is relatively simple – certainly far simpler than demodulating and decoding it!

Signal Design

My job here is to carry packets or frames of arbitrary data from the ARISSat-1 experiments and onboard computer (IHU) to a ground station where they can be decoded and displayed by a computer running special software. This paper does not define or describe the actual data to be carried on this link, so my description here starts with the framing layer and works down to the modulation used on the physical channel. Consult the other papers on ARISSat-1 for information on the contents of the frames actually transmitted during flight.

BPSK1000 encoding can be succinctly described as follows:

- HDLC framing with a 32-bit CRC;
- constraint length 7, rate 1/2 convolutional forward error correction (FEC) coding;
- 128-way convolutional interleaving with "bit reversed" delay line ordering; and
- DBPSK (Differential Binary Phase Shift Keying) modulation.

These elements are all "off the shelf", well documented in the textbooks and used commercially for many years. What follows is a more detailed description and my rationale for selecting each one for this project.

Framing

When I started work on this format, the telemetry data formats and sizes were not yet established. Rather than pick a frame size that would almost certainly turn out to be wrong, I chose to support variable length framing for its versatility.

One of the best known and most widely used methods of variable length framing on synchronous bitstreams is HDLC, familiar to most hams from its use in the AX.25 packet radio protocol. Stripped to its bare essentials, an HDLC frame in BPSK1000 is shown in Figure 1.

The special 'flag' sequence 01111110 represents the end of one frame and the start of another. (Two frames may share a common flag or there may be extra flags between frames - the decoder will work either way.) As called for in the HDLC standard, user data is transmitted least

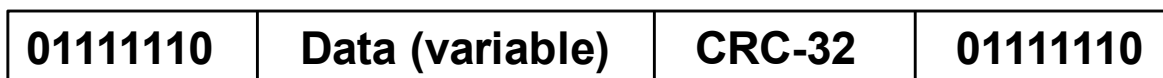


Figure 1: HDLC frame.



significant bit first between the opening flag and the CRC.

To ensure transparency, the HDLC encoder inserts a '0' bit whenever five contiguous '1' bits occur in the data stream. When the receiver sees five '1' bits followed by a '0', it automatically removes the '0'. Although bit stuffing can consume up to 16.7% of the channel capacity in overhead, this occurs only for a data stream of all 1's. The overhead for bit stuffing on random data is much less.

I made only one significant change to standard HDLC framing as used in BPSK1000: I replaced the usual 16-bit CRC with a 32-bit CRC with the generator polynomial: [2]

$$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1.$$

The stronger CRC greatly reduces the chance that a random bit stream will decode as a valid frame.

BPSK1000 treats the contents of the "data" field as opaque, i.e., it does not care about their contents. Although ARISat-1 doesn't send AX.25 headers, nothing prevents another BPSK1000 application from doing so. I.e., BPSK1000 could easily carry standard AX.25 packet radio data, although for reasons discussed later it is not well suited to the half duplex, shared channel mode of operation of most amateur packet radio channels.

The fact that BPSK1000 does not assume the presence of AX.25 headers is one reason I decided to enlarge the CRC. Spurious HDLC frames are common in packet radio but they are easy to detect by their absence of an AX.25 header. Because I did not want to make any assumptions about the contents of ARISat-1 HDLC frames, I chose a longer CRC that all but eliminates spurious frames in the first place.

In sum, HDLC converts an asynchronous stream of arbitrary length frames into a continuous 500 bps bit stream for the FEC encoder. If there's nothing to send, the HDLC encoder must emit an idle flag stream. The HDLC decoder at the receiver regenerates the original series of frames with the 32-bit CRC providing strong error detection. Although BPSK1000 includes forward error correction (FEC), the HDLC CRC is necessary because a Viterbi decoder still makes errors when its error correction ability is exceeded and it cannot reliably signal when they occur.

Convolutional Error Correction Coding

To allow for the correction of a limited number of channel errors, the HDLC-encoded bit stream is convolutionally encoded with the CCSDS standard constraint length 7, rate $\frac{1}{2}$ convolutional code using connection vectors $G_1=1111001$ binary (171 octal) and $G_2=1011011$ binary (133 octal): [3] (See Figure 2) (<http://public.ccsds.org/publications/archive/131x0b1.pdf>)

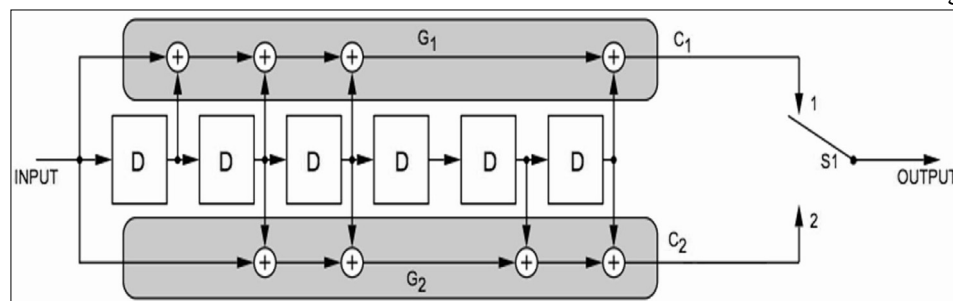


Figure 2: HDLC encoder.

For each data bit from the HDLC encoder, the convolutional encoder generates two symbols, C1 and then C2. I use the CCSDS convention for the ordering of the two encoded symbols but I do not invert either symbol.

The CCSDS and JPL versions of this encoder both invert the output of the 133 (octal) XOR ladder (shown here as symbol C2). Their reason for doing so is to limit the maximum possible number of contiguous 0's or 1's that can come out of the encoder regardless of the data so as to guarantee a minimum symbol transition density for demodulator tracking. That is unnecessary here because HDLC bit-stuffing ensures that no more than five consecutive '1' bits can ever occur in user data, and an HDLC flag consists of six consecutive '1' bits. An HDLC abort, if ever generated, need (and should) consist of only seven '1' bits, followed by an idle flag stream until more data is available for transmission. [4]

Although HDLC does not limit runs of '0' bits, this is not necessary here because of the use of differential encoding before BPSK modulation that turns every '0' bit into a 180° carrier phase transition. It is only necessary to break up consecutive '1' bits as differential encoding sends them as no change in carrier phase.

Convolutional Interleaving

Viterbi decoders tolerate burst errors poorly. Since a major requirement for ARISat-1 is to mitigate the deep fading common with

unstabilized satellites in low earth orbit, an interleaver is necessary.

Interleaving scatters the encoded symbols in time before transmission and rearranges them into their original sequence at the receiver before decoding. This has the effect of breaking up a burst of errors into a much longer sequence of single bit errors that are much more easily corrected.

There are two major classes of interleavers: block and convolutional. Block interleaving

is straightforward. The encoded symbols are written into an array by rows and read out by columns for transmission. The receiver reverses the process by writing the array by columns and reading it out by rows.

Delay can be a problem with block interleaving, especially when a large block is required to tolerate deep fades. Nearly the entire block has to be written before transmission can start, and then the entire block has to be received before decoding can even begin. Block interleaving usually requires that a single block size be selected in advance, though some systems use several sizes. However, this doesn't necessarily restrict the system to a fixed data frame size; HDLC or something similar could support variable length data frames just as it does here.

These drawbacks are not necessarily serious. For example, in its FEC mode the AO-40 IHU generated fixed sized 256-byte data blocks more or less instantly, so the delay from block generation in software to the decoded block being available on the ground was equal to just one block transmission time (13 sec) plus the times needed for the IHU to encode it and the ground CPU to decode it.

But when the source is a steady bit stream, or a series of small data frames spread out in time, the encoder must wait for the interleaver block to fill before it can even begin to transmit it. The last data frame in the block will not see any difference, but this can nearly double the time between the

generation of the first (small) data frame in the block and when it is available on the ground. [5]

An alternative to block interleaving is convolutional interleaving, [6] and I chose it for BPSK1000 to help reduce latency when small HDLC frames are carried. Just as a convolutional encoder differs from a block encoder by operating on a continuous data stream rather than fixed-sized blocks, a convolutional interleaver also operates on a continuous stream. This is well suited to the continuous stream of encoded symbols from the convolutional encoder.

Here is a block diagram of a small (order 4) convolutional interleaver (Figure 3):

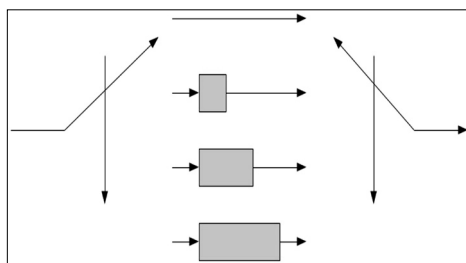


Figure 3

The symbol stream to be transmitted is fed in on the left to the 1:4 rotary switch that deposits each symbol in turn into one of four delay lines whose lengths range from 0 in the first row (no delay) to 3 time units in the last row. Their outputs are collected by the 4:1 rotary switch operating synchronously with the input switch. The delay lines clock only when selected by the rotary switches, which operate synchronously, and the output of the right hand rotary switch goes to the transmitter.

At the receiver, the incoming symbol stream passes through the deinterleaver, the mirror image of the interleaver (Figure 4):

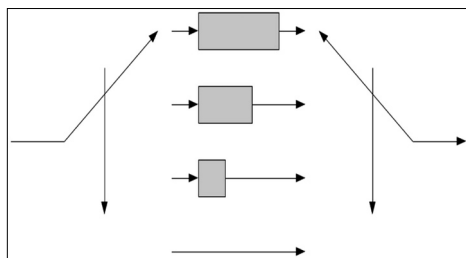


Figure 4

The essential feature of convolutional interleaving is that in each row, the length of the interleaver delay line plus the length of the deinterleaver delay line is a constant, N . In this example, $N = 3$. So the overall effect of the interleaver and deinterleaver combined is to reproduce the original input stream, in its original sequence, delayed

by a number of time units equal to N times the number of rows R ; in this example, that would be 12 time units.

N can actually be any number you choose, though it can't be less than $R-1$ if each row is to have a delay on each side of the channel different from all the other rows. Otherwise some input symbols will be sent in their original order, and that defeats the point of interleaving. And if you make N bigger than this, you're just adding unnecessary delay. So for all practical purposes, $N = R-1$.

The main advantage of convolutional interleaving over block interleaving is that less memory and less end-to-end delay are required for a given amount of fade protection. One can think of convolutional interleaving as being something like block interleaving but with a single block split between transmitter and receiver.

Convolutional interleaving also has its drawbacks. Because there's no well defined start or finish to the data stream as in block interleaving, it is not well suited to short, bursty transmissions like those on a shared, multiple access channel (e.g., conventional packet radio). Conversely, a system that uses block interleaving could key up, send a single block and stop transmitting without having to allow any time to "prime" the deinterleaver or to flush the interleaver.

But that doesn't matter for our application because it's a one-way continuous broadcast. The ground station will still need to prime its de-interleaver at the beginning of the pass. At first this might seem a disadvantage relative to block interleaving, but it's not. Satellite AOS can occur at any time, and were it to occur in the middle of a block in a block-interleaved system you'd have to wait up to one block time for the first full block to start. With convolutional interleaving, priming of the de-interleaver takes the same amount of time no matter when you start.

Bit-reversal convolutional interleaving

In most convolutional interleavers, the delay in each row is monotonically increasing (and decreasing at the deinterleaver, or vice versa). But this isn't mandatory. Just as the rows and columns of a block interleaver can be written in any order (even random) so long as the receiver mirrors the transmitter and puts everything back in its original place, anything goes.

This got me thinking about playing with some alternative orderings. One common

trick in block interleaving is to access the rows and columns in "bit reversed" order. Take the integers from 0-7 and write them in binary:

000, 001, 010, 011, 100, 101, 110, 111

Now reverse them left-right and convert back to decimal:

0, 4, 2, 6, 1, 5, 3, 7

Now access the rows or columns in that permuted order.

Why do this? Well, it helps to spread out adjacent input symbols during an error burst (e.g., a fade). The first errored symbols in the burst are as far apart as they can get in the memory available. Additional errors are evenly sprinkled between the previous errors, though of course if the burst lasts long enough it will eventually have to fill in all the gaps and decoding failures will occur.

Bit reversal interleaving requires that the number of rows be a power of two, so I chose an interleaving order of 128. This implies that each row contains a total delay of 128 bit times divided between transmitter and receiver for an end-to-end interleaving/deinterleaving delay of 16,384 symbols. [7]

Modulation

BPSK? QPSK?

The power efficiency requirement pretty much mandates either BPSK or QPSK. In theory, QPSK (quadrature phase shift keying) requires the same power as BPSK and only half the bandwidth. In practice it is difficult to make QPSK perform as well as BPSK especially when the data rate is low compared to the frequency uncertainty, as is the case here, because QPSK has much tighter receiver requirements than BPSK for carrier frequency and phase tracking. So despite the bandwidth limitations of SSB filters I decided to stick with BPSK. It has a very long history on Amateur Radio satellites and satellites in general. [8]

The requirement that the signal fit a SSB receiver means that the modulated bandwidth cannot exceed a typical SSB voice filter, usually about 2400 Hz. It should be somewhat narrower to allow for errors in tuning and to stay away from the nonlinear group delay distortion typically found at the edges of crystal filters designed for voice operation where such delay isn't a serious problem.

Signaling rate

The Nyquist bandwidth of BPSK is 1 bit/

sec/Hz. That is, I could in principle fit 2,000 bits/sec into 2 kHz of a SSB passband while leaving 200 Hz on either side. But achieving 1 bit/sec/Hz with BPSK requires very tight filtering with very little tolerance of symbol timing errors and group delay distortion. The FIR filter used at both transmitter and receiver must be quite long to achieve the necessary response, and processing time in the onboard DSP is limited.

Tightly filtered BPSK also generates a fair bit of overshoot in the RF envelope. This rules out efficient constant-envelope amplification, although that's not a problem in ARISSat-1 because it already gives us a linear amplifier for the entire 2 m downlink including the telemetry beacon.

So to be conservative, I selected a BPSK symbol rate of only 1 kHz (i.e., 1000 symbols/sec) with a raised-cosine filter having 100% excess bandwidth, i.e., 1/2 bits/sec/Hz. This is achieved with a finite-impulse-response (FIR) filter with 454 taps running at a sample rate of 48 kHz; the same filter taps are used as the matched filter in the demodulator on the ground. Although the resulting bandwidth is still 2 kHz, instead of being flat the signal spectrum rolls off from a peak at the nominal center frequency to nulls at ± 1 kHz, with relatively little energy near the nulls.

Manual tuning with the CW beacon

As mentioned in the requirements section, the CW beacon is placed in the lower null of the telemetry beacon spectrum, 1 kHz below the nominal carrier frequency of the BPSK signal. All the operator need do is tune the receiver in SUB mode so that the CW beacon comes out at 500 Hz, and this will automatically center the BPSK signal in a typical SSB receiver bandwidth; i.e., the signal will extend from the lower null at 500 Hz to an upper null at 2500 Hz with the carrier at 1500 Hz, the nominal center of the SSB passband.

Why not coherent BPSK?

Because our spacecraft will transmit on 2 m from low earth orbit, path losses will be relatively low and the signal will, on average, be fairly strong. But experience has shown that unstabilized spacecraft in low earth orbit often exhibit deep, unpredictable fading. That's the reason for the deep interleaver.

Fading (or its absence) is also a major consideration in the design of BPSK modems as it can be implemented in two very different ways depending on which

channel impairment is more important: thermal (Gaussian) noise or fading.

Satellite links are traditionally designed to minimize thermal noise, so most use a form of PSK that requires the receiver to form a clean local copy of the unmodulated signal carrier to serve as a phase reference for the receive signal. This is coherent demodulation.

This can be done in one of two ways. The first is for the transmitter to send a residual carrier along with the data so that the receiver can track it with a conventional phase lock loop (PLL). The second is for the receiver to reconstruct the carrier entirely from the data by stripping off the modulation.

Although residual carriers are still used in some deep space links, particularly at the extremely low data rates sent by a spacecraft in "safe mode", this takes power away from the data. It is much more common to suppress the carrier entirely at the transmitter so that all of the power can go into the data. The receiver recovers the carrier from the data with one of two mathematically equivalent methods: the squaring loop and the Costas loop.

Especially when strong FEC is used, these loops can suffer significantly from "squaring loss", an effect that reduces the signal-to-noise ratio of the recovered carrier. This happens because the raw symbols (before FEC decoding) are very weak, so when the loop strips off the modulation it does so with a fairly high error rate that degrades the recovered carrier.

Squaring loss can be overcome by merely narrowing the loop filter, effectively averaging out the noise over a longer interval that may span hundreds or even thousands of channel symbols. This is fine if the signal is very stable, as on most deep space links. But it can become completely unworkable when the signal is subject to rapid fading, e.g., on ionospheric or land mobile channels, especially at the low symbol rates we're using here.

We also have fading problems with our low earth orbiting spacecraft. Although we may have continuous line of sight, the spacecraft may tumble unpredictably and sweep antenna nulls past the receiver, causing deep fades accompanied by a sudden reversal of signal phase that would defeat a conventional coherent BPSK demodulator of the Costas or squaring loop type.

Differential BPSK

For this reason I have chosen a form of BPSK for ARISSat-1 that doesn't require the receiver to form a carrier phase reference. This is DBPSK, differential binary phase shift keying. It is generated just like ordinary BPSK with one crucial exception: before it modulates the transmitter, the data stream is differentially encoded so that a binary zero is transmitted as a 180° change in the phase of the transmitted carrier while a binary one is transmitted as no change from its previous state. This compares with conventional BPSK where a binary zero corresponds to the RF carrier with a 0 degree phase shift and a binary one corresponds to the RF carrier shifted by 180° .

The advantage of DBPSK for our application can be seen in how it is demodulated. The receiver can simply compare adjacent incoming symbols, looking for sudden 180° changes in phase - whatever it may be.

This is extremely easy to implement. In software (the only way to build modems nowadays!), the standard way to demodulate DBPSK is to first convert the input signal to complex baseband at zero frequency. That is, we generate a complex sinusoid (a matching sine and cosine wave) at the estimated signal frequency and multiply it by the input signal (which can be either complex or real) to produce a complex baseband signal at (approximately) zero frequency. If our frequency estimate is correct, an unmodulated carrier becomes a stationary phaser at some arbitrary phase angle (since we don't know or care about its phase). Should the signal flip phase by 180° , this phaser will rotate 180° . By computing the dot product of pairs of complex symbols we simultaneously demodulate the data and remove the differential encoding; this works because the dot product of a vector with itself is a positive number while the dot product of a vector with itself flipped 180° is a negative number.

So instead of generating a nice clean phase reference by averaging some hundreds of symbols, in DBPSK demodulation we just use each symbol as the phase reference for the next. The dot product yields a scalar (a number) that tells us how certain the symbol estimate is, just what we need for our soft-decision Viterbi decoder.

Effects of frequency errors on DBPSK demodulation

What is the effect of an error in the frequency

estimate? If the local oscillator doesn't match the incoming signal frequency, then the baseband phaser will rotate clockwise or counterclockwise depending on whether the oscillator is above or below the actual signal frequency. But if the error is small, the vector will rotate very slowly. It may still rotate many times in a second, but because we're only comparing adjacent symbols – not trying to form a reference from hundreds of them – it only matters that it rotate no more than a few degrees during one symbol time.

Let's say the BPSK is at a symbol rate of 1 kHz and our frequency estimate is off by 100 Hz. The baseband phaser will rotate 100 times per second, but that's only 1/10 of a rotation (36°) from one symbol to the next. Because the magnitude of the dot product of two vectors is proportional to the cosine of the angle between them, a small frequency error decreases the magnitude of the dot product to $\cos(36^\circ) = 0.81$. This corresponds to a loss of $20 \cdot \log_{10}(0.81) = -1.84$ dB. If the frequency error can be kept to 50 Hz, then the loss is only $20 \cdot \log_{10}(\cos(18^\circ)) = -0.44$ dB, a tolerable figure.

Decoding BPSK1000

Nearly every digital format is easier to generate than it is to decode and BPSK1000 is no exception. This is because the receiver has to first determine various signal parameters before it can demodulate and decode the signal and this searching and tracking consumes most of the CPU time.

Input format

My BPSK1000 demodulator accepts a 16-bit linear PCM audio signal sampled at 48 kHz. The sample rate is a compile-time constant; although it can be changed, the structure of the code requires it to be an integer multiple of the symbol rate (1 kHz) so if you have a signal at some other sample rate (e.g., 44.1 kHz) it's easier to simply convert it with a tool like *sox*.

Chunking

My demodulator operates on blocks of PCM samples called "chunks". The chunk size, currently 512 symbols (512 ms), is chosen to contain enough energy for the carrier frequency and symbol timing search while being short enough that these parameters remain essentially constant within the chunk.

Carrier frequency and symbol timing search

The first step in decoding BPSK1000 is to

determine the carrier frequency and symbol timing. Because it's DBPSK, an estimated frequency will do; as shown above, ± 50 Hz will do, so I search in 100 Hz steps. I must also determine where each symbol starts and stops. So I run a series of trial demodulations with every possible combination of carrier frequency (in 100 Hz steps) and symbol time. Since the sample rate is 48 times the symbol rate, this means trying 48 different hypotheses for symbol timing.

No attempt to acquire data is made at this stage. Only the total demodulated energy is calculated and the carrier frequency and symbol timing parameters that yield the largest demodulated energy are taken as the best estimates. This step is both CPU intensive and parallelizable, so it's a natural for a machine with multiple CPU cores. But in practice this search is actually pretty fast except on very old and slow machines thanks to a fast FIR filter that uses vector instructions.

Carrier frequency refinement

After the initial estimate, the carrier frequency estimate is refined by simple linear extrapolation. As explained above, when a frequency error is present the baseband signal vector slowly rotates. This rotation can be detected by summing the magnitudes of the vector cross products of adjacent pairs of baseband phasors. When those vector cross products are zero (except for noise) the frequency error is zero. This step is actually overkill since, as stated, 50 Hz frequency errors have little effect on demodulation. I do it mainly to give the user a more precise frequency reading.

Tracking and demodulation

Once the carrier frequency and symbol timing are determined, there's no need to repeat the full-blown search unless the signal is lost. Changes in frequency and timing between each chunk are tracked by simply running a trial demodulation (and looking at the demodulated energy) at values bracketing the current estimates.

Then the data can finally be demodulated.

Deinterleaving and decoding

Convolutional interleaving, as described above, assumes that the rotary switches in the interleaver and deinterleaver are synchronized. If not, the deinterleaver produces garbage.

There are no sync patterns in the BPSK1000 format to indicate interleaver phasing. Correct interleaver phase has to be found

by brute force and this automatically synchronizes the Viterbi decoder phase as well. Once the DBPSK demodulator has acquired and demodulated a symbol stream, the soft decision samples are scaled and fed to 128 virtual deinterleavers, Viterbi decoders and HDLC decoders, one for every possible interleaver phase. Eventually, one of them will produce a HDLC frame with good CRC. Now that we know the correct phasing, the other 127 virtual decoders can be shut down. They remain shut down as long as we continue to get valid HDLC frames. If too much time passes without a valid frame, then all 128 decoders are re-enabled until another frame is decoded.

This was my main motive for using a 32-bit CRC with HDLC; I was concerned that spurious frames would occur frequently with the 16-bit CRC and interfere with the determination of the correct deinterleaver phase.

This brute force approach isn't nearly as bad as it sounds; even when all 128 decoders are running, DBPSK demodulation still takes about half of the program's total CPU time. Demodulation and decoding run in separate threads, both for architectural convenience and to take advantage of extra CPU cores.

Concluding thoughts

After the launch of AMSAT-OSCAR-40, I designed and implemented an experimental FEC-coded telemetry format (<http://www.ka9q.net/ao40/>) designed to deal with the deep periodic fading characteristic of the S2 transmitting antenna at high squint angles.

This project was very successful, though it was eventually cut short by the complete failure of the spacecraft. I learned a great deal from that experience, and while the ARISSat project has a different set of design objectives and constraints, I was able to reuse many important design elements.

Yet ARISSat-1 will be a very different spacecraft than AO-40, if only because it will operate from LEO while AO-40 was in a high orbit. BPSK1000 is also rather different from my AO-40 FEC design, partly because the latter was constrained to work with the existing Phase III IHU hardware and software: the symbol rate had to be 400 Hz, the modulation had to be BPSK with Manchester (biphase) encoding, the data was fixed-length 256-byte blocks, and so on.

I had none of those constraints with ARISSat-1. In some ways that lack of constraints actually made BPSK1000 more difficult to



design, but I think the result is better for it. Although I've used mostly "off the shelf" modem design elements, as a system BPSK1000 is a new design that has never flown on an Amateur Radio satellite. I would have liked to implement a range of operating modes, e.g., data rates, bandwidths and interleaver depths, selectable by command so that the optimal mode for the actual conditions could be selected in flight. This wasn't possible so I designed a single mode to be as robust as possible, e.g., by picking a conservatively user low data rate. I very much look forward to seeing it operate from space so we can gain the experience necessary to design improved modems for future flights. With digital signal processing both on the ground and on the spacecraft, and with the new generation of software defined radios, the potential for new amateur satellite digital modes is extremely bright.

Footnotes

- [1] Clarke's Law (after Arthur C. Clarke) states that any sufficiently advanced technology is indistinguishable from magic. My corollary to Clarke's Law states that any sufficiently advanced communication scheme is indistinguishable from white noise.
- [2] For the 32-bit CRC polynomial I chose the one used in Ethernet, sometimes known as CRC-32.
- [3] The polynomials and connection vectors for code generator polynomials can be written with either the most or least significant term first, often leading to considerable confusion. The convention here is taken from JPL Deep Space Network Handbook 810-005, 208A, page 12, in which the leftmost bit in the connection vector represents the taps encountered by the newest data bit entering the encoder. Some other references, plus my own software, use the opposite convention. This makes the vectors 1001111 and 1101101 binary; 117 and 155 octal, or 6D and 4F hex.
- [4] The interleaver complicates this issue considerably. Even with HDLC bit stuffing (which I use) and alternate FEC symbol inversion (which I don't use) I can conceive of pathological cases where the interleaver might produce a very long string of 1's from a sequence with a bounded number of them. I could have added a scrambler, as I did for AO-40, but with the addition of HDLC bit

stuffing I just didn't think the remaining problem was serious enough to warrant it here.

- [5] As an analogy, consider a group of shuttle busses that run on demand. Each bus carries 50 people and the drivers like to fill all the seats. If you're part of a group of 50 people who arrive at the bus stop together, you can quickly board the first bus and leave right away. But if 50 people arrive individually over the space of 15 minutes, the first person to board the bus will have to wait 15 minutes for the bus to fill before it leaves.
- [6] Not to be confused with convolutional coding, which I'm also using here!
- [7] Strictly speaking the delay should be $127 \times 128 = 16,256$ symbols but it the code was easier to write with a little more delay.
- [8] Some amateur satellites using BPSK include the Phase 3 series (Phase 3A and Oscar 10, 13 and 40) and the Pacsat series, e.g., Oscar 16.

References

HDLC framing is described in many places. Just a few include:

- AX.25 Amateur Packet Radio Link Layer Protocol Version 2.2, 1998, <http://www.tapir.org/pdf/AX25.2.2.pdf>
- International Telecommunications Union X.25, Interface between Data Terminal Equipment (DTE) and Data Circuit-terminating Equipment (DCE) for terminals operating in the packet mode and connected to public data networks by dedicated circuit, <http://www.itu.int/rec/T-REC-X.25/en>
- International Standard ISO/IEC 13239, "Telecommunications and information exchange between systems - High-level data link control (HDLC) procedures", http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=37010.

The $k=7$ $r=1/2$ convolutional code is described in many places including:

- DSN Telecommunications Link Design Handbook, 208, Rev. A Telemetry Data Decoding, May 18, 2009, <http://deepspace.jpl.nasa.gov/dsndocs/810-005/208/208A.pdf>
- CCSDS 131.0-B-1, TM Synchronization and Channel Coding, Blue Book Issue

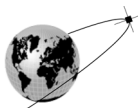
1, September 2003. <http://public.ccsds.org/publications/archive/131x0b1.pdf>

- FEC Encoding for AO-40 Telemetry, including links to my published paper in the 2002 AMSAT Annual Meeting. (<http://www.ka9q.net/ao40/>) 

The AMSAT Store (see www.amsat.org) is back online! Here is a new AMSAT product. A handsome and functional AMSAT 10' locking carpenter's tape with both inch and metric scales.

Minimum donation: \$5





would have to be flown on Baikonour on 11 Jan, meaning that all testing and integration would need to be finished by 31 Dec 10 before the RSC-Energia shutdown.

As I'm writing this column, Lou and Mark made their trip to Russia and are preparing to return home, not having seen the spacecraft as it still sits in Customs. This means that the integration/testing that must take place by 31 Dec 10 will be done without AMSAT direct participation. Consequently, Lou and Gould Smith, WA4SXM, have been drafting detailed testing procedures that will be shared with the Russians to give them guidance on ARISSat-1 testing to commence after the satellite is delivered to RSC-Energia. These detailed instructions will be translated into Russian. This is a significant challenge as the procedures must include appropriate detail for someone who has never seen the ARISSat-1 package. Much of the corporate knowledge that Lou and Gould share must be written down and explained in such a way that the testing can be accomplished by capable individuals who have no prior experience with this equipment.

Given the hang-ups with Russian Customs, there is a real possibility as I'm writing this note that ARISSat-1 will not be ready for shipment on Progress 41P. If there is a delay, I am still confident that ARISSat-1 will be deployed sometime in 2011. While we are all disappointed by the turn of events, we are encouraged by the warm reception that Lou and Mark received from RSC-Energia. While in Russia, they took full advantage of the opportunity to brief RSC-Energia personnel on ARISSat-1, met with officials of Kursk State Technical University (who developed the experiment for ARISSat-1/RadioSkaf-V), and established the appropriate contacts so that if there are issues raised during integration/testing both RSC-Energia and AMSAT personnel know who to reach to resolve any issues.

On a different note, Project Fox is moving along well under the leadership of VP-Engineering Tony Monteiro, AA2TX. An initial systems requirements document has been drafted and has gone through several revisions. See Tony's article on Project Fox in this issue of *The AMSAT Journal* for more details.

In addition, a conference call was held on

20 Dec 10 with engineering students of SUNY-Binghamton who are working on the AMSAT Next Gen project as part of their senior level course work. Students provided their initial design thoughts (Preliminary Design Review) for a number of CubeSat design areas, including solar panel and antenna deployment, power management, attitude determination and control, command and data handling, spacecraft physical structure and thermal model, and spacecraft systems integration. A number of AMSAT engineering personnel participated, including Tony Monteiro, AA2TX; Dick Jansson, KD1K; Bob Davis, KF4KSS; Bill Reed, NX5R; and Bryan Klofas, KF6ZEO. These volunteers will review what the students have developed to date and provide feedback on their design work. If all goes well, there will be a model of their work on display at the AMSAT booth at the 2011 Dayton Hamvention. These students are working on some very interesting designs and it is our hope that their work may eventually be adapted to an AMSAT spacecraft. This also provides an opportunity to help develop the next generation of spacecraft builders.

Best wishes during the holiday season. Many thanks to all AMSAT members for your support of AMSAT and I look forward to working with you in 2011.

73,

Barry A. Baines, WD4ASW

President



From ANS:

Henk, PA3GUO, has posted a YouTube video (HD) that features the O/OREOS satellite from launch to telemetry reception and decode, including DK3WN's Software and Mission control upload: <http://www.youtube.com/watch?v=Qh6IH5vqw-Y>

Henk also has a video of the 'bare' telemetry stream at: <http://www.youtube.com/watch?v=M4Ka-RVpEHU>

O/OREOS audio recording: http://www.pa3guo.com/OOREOS_PA3GUO.mp3

FASTRAC/FAST1 audio recording: http://www.pa3guo.com/FAST1_PA3GUO.mp3

(Henk says you can use MixW with your MP3 your player to decode the telemetry.)

AMSAT-ZL's Fred Kennedy, ZL1BYP, says the KiwiSAT team is in need of a small quantity supply source of space quality thermally conductive epoxy for KiwiSAT. They will need probably less than 10 grams total over the last few months of KiwiSAT construction. Minimum order quantities have been quoted as 100 tubes and this amount is not needed nor can be afforded. Contact Fred at: fredk@kcbbbs.gen.nz if you can help.

For those who collect archived publications here is a link to a recently posted digitized copy of the 1985 1296/2304 MHz Conference, which has evolved into the Microwave Update MUD Conference. Thanks to the work of Brian, WA1ZMS, these proceedings have been digitized and can be downloaded in 7 sections from <http://www.wa5vjb.com> in the Reference Section.

Captain Yuri Bodrov, Master of the M/V "MOTTLER", says he will be operating as UT1FG/MM on AO-7, AO-27, AO-51, HO-68, SO-50, SO-67, and VO-52 from the wet squares in the ocean on his next voyage around South America from Brasil/Paranagua to Argentina/San-Lorenzo, through the Strait of Magellan, Chile/San-Antonio/Punta Potache, and the Panama Canal. 



AMSAT[®] President's Club!

Join Now!

Contribute to **AMSAT[®]** directly through easy, automatic charges to your credit card.
Donations may be USA tax deductible. (Check with your tax advisor.)

Titanium Donors contribute at least US \$400 per month.

Every **Titanium Donor** receives gifts of:

- ✓ Everything at the Platinum Donor Level, PLUS:
- ✓ A very special, *hand crafted*, AMSAT President's Club Donor's pin.

☐ \$400 / month

☐ \$4800 one time

Platinum Donors contribute at least US \$200 per month.

Every **Platinum Donor** receives gifts of:

- ✓ Everything at the Gold Donor Level, PLUS:
- ✓ A very special, *hand crafted*, AMSAT President's Club Donor's pin.

☐ \$200 / month

☐ \$2400 one time

Gold Donors contribute at least US \$100 per month.

Every **Gold Donor** receives gifts of:

- ✓ Everything at the Silver Donor Level, PLUS:
- ✓ A personal invitation to the AMSAT Dayton dinner (free).

☐ \$100 / month

☐ \$1200 one time

Silver Donors contribute at least US \$50 per month.

Every **Silver Donor** receives gifts of:

- ✓ Everything at the Bronze Donor Level, PLUS:
- ✓ A personal invitation to a reception with the AMSAT Directors, Officers and project designers and builders (free).
- ✓ Annual endorsement sticker after each year of donation.

☐ \$50 / month

☐ \$600 one time

Bronze Donors contribute at least US \$25 per month.

Every **Bronze Donor** receives gifts of:

- ✓ A unique "AMSAT President's Club" certificate.
- ✓ A specially designed AMSAT President's Club Donor pin.

☐ \$25 / month

☐ \$300 one time

Core Donors contribute at least US \$10 per month.

Every **Core Donor** receives gifts of:

- ✓ A specially designed AMSAT President's Club Donor pin.

☐ \$10 / month

☐ \$120 one time

Choose where your contribution should go:

☐ AO-Echo ☐ AO-Eagle ☐ ARISS ☐ Future Projects ☐ General Fund

☐ RENEW my
President's Club
AUTOMATICALLY
every year!

☐ I prefer to make my donation in one yearly payment of \$ ____.

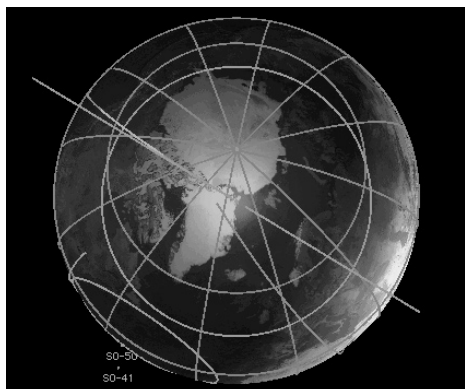
(Credit card only, please.)

THANK YOU!



MacDopplerPRO X

The premier Satellite tracking and station automation application for the Macintosh - OS 9 & OS X



MacDoppler for Cocoa gives you a seat right in the heart of the Operations & Command Centre for every satellite in orbit, providing any level of station automation you need from assisted Doppler Tuning and Antenna Pointing right on up to a fully automated Satellite Gateway !

It will calculate the position and relative velocity of the satellites you are tracking and automatically adjust the Doppler shift on both transmit and receive as well as pointing your antennas with predictive dead spot crossing so that a pass is never interrupted.

A Universal Binary that runs native on Intel and PPC Macs and provides separate panels for the map (2D or 3D), the radio and rotor controls, a sorted table of upcoming satellite passes and a Horizon panel that graphs upcoming passes as a function of elevation over time.

Now available from AMSAT at a special member discount donation!

martha@amsat.org
850 Sligo Ave.,
Silver Spring MD,
20910-4703 USA.
(301) 589-6062, (301) 608-3410 (Fax)

Dog Park Software Ltd.
www.dogparksoftware.com

Do you still use manual tuning to remove Doppler changes? Why not automate? **Uni-Trac 2000** will automatically tune your Receiver (and Transmitter) to compensate Doppler frequency changes. Uni-Trac will also **STEER** your Antenna !

Uni-Trac 2000, including Interface Cable \$280.00
FOB Concord, Mass. 01742, USA.



Spectrum International, Inc.,
P.O. Box 1084,
Concord, Mass. 01742, USA.
Tel: (978) 263-2145
Fax: (978) 263-7008

Help Wanted AMSAT needs qualified volunteers for a number of positions!

If you want to be a part of the solution in making AMSAT operational and work toward designing, building and finding a launch, WE NEED YOU!

No pay for 5+ hours per week, but a great deal of satisfaction in knowing you are helping make something happen.

We need people with these skills:

- Marketing
- Sales
- Software engineering
- Hardware engineering
- PC based software design
- Web design and maintenance
- Project management
- Documentation
- Hardware testing
- Software testing
- Hardware prototyping
- Specific area Web site information maintenance
- Video recording, digital conversion and editing
- Write technical or instructional material
- Educational activities
- Promoting AMSAT as an Area Coordinator



and any other area you think you think you can make a difference.

We often have people volunteer, but many don't deliver on their promises. So, pick an area that you think needs improvement and explain what you will do to make it better. Then do something to show that you can follow through.

Send your suggestions, actions plans and a description of why you are capable of doing this task to Gould Smith, WA4SXM, wa4sxm@amsat.org



2010 Space Symposium and AMSAT-NA Annual Meeting

Held October 8-9-10, 2010 at Chicago/Elk Grove Illinois

Contents include articles on AMSAT's Satellites, Opportunities in Education, Satellite and Earth Station Technology and more. For example:

- Software Technology on ARISSat-1
by Tony Monteiro, AA2TX
- AMSAT-Fox Preview
by Tony Monteiro, AA2TX
- AMSAT NextGen CubeSat Proposal - Evolutionary Leveraging of Assets and Opportunities
by NextGen Satellite Team - Alex Harvilchuck, N3NP
- Ham TV Experiment On-board ISS
by F Azzarello, IW8QKU, E. D'Andria, IOELE, T. Giagnacovo, IZ0LTG, and P. Tognolatti IOKPT
- BPSK-1000 Format for ARISSat-1
by Phil Karn, KA9Q
- How to Work the FM Satellites With Your Handheld Radio
Clint Bradford, K6LCS

Now Available!

To Order: Proceedings - 2010

(8½" x 11", 272 pages)

See the AMSAT Store: www.amsat.org or

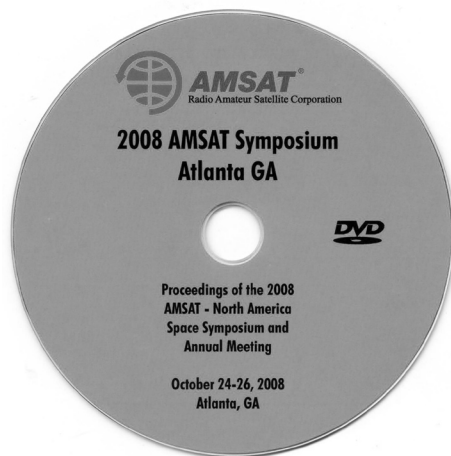
Write, telephone or FAX:

AMSAT-NA
850 Sligo Ave. #600
Silver Spring, MD 20910, USA

Tel: 301-589-6062
Fax: 301-608-3410

Minimum Donation:

\$25 anywhere + postage.
For details click on "AMSAT Store" at www.amsat.org. VISA, MasterCard or Discover charges accepted.



2008 Proceedings DVD's also available. Contact the AMSAT Office.

A new DVD for the 2008 Symposium is available. It has the presentations, audio files and papers.
Dan Schultz's Banquet Talk on The Hubble Space Telescope Repair Mission - slides and audio
Phil Hartman's Introduction to DSP Workshop - slides and audio plus
2004 Symposium papers
2005 Symposium papers
2007 Symposium papers





ARISSatTLM - A Software Demodulator and Telemetry Display Program for the ARISSat-1 Satellite

Douglas D. Quagliana, KA2UPW/5, dquagliana@aol.com

Abstract

ARISSatTLM is a complete software DSP demodulator and telemetry display program for the telemetry from the ARISSat-1 satellite.

Introduction

ARISSatTLM is a graphical Windows program for displaying telemetry from the ARISSat-1 satellite (formerly known as SuitSat-2). The program is written in C, compiles with gcc and is freely available under the terms of the GNU Public License (GPL).

ARISSatTLM includes C source code for the sound card routines originally written by Tony Monteiro, AA2TX, and a BPSK1000 software modem with digital signal processing source code originally written by Phil Karn, KA9Q. [1] [2] Both were modified for use in ARISSatTLM. Karn's code was originally written for gcc on an Apple Mac computer and required changes to compile under a similar version of the Cygwin gcc compiler for Windows. Similarly, Monteiro's code was written for a PC, but not written for gcc and only minor changes were needed. ARISSatTLM provides some "glue," in the form of C code, to get the pieces to talk together. It also includes original code to display the decoded telemetry values. The software runs in real time on a 1 GHz Pentium III PC. It should be able to process prerecorded WAVE files on even slower machines.

ARISSatTLM configures the sound card to produce digital samples at 48,000 samples per second with 16 bits per sample. This is the format that the BPSK1000 software modem expects. The bytes arrive from the sound card routines and are passed to the BPSK1000 software modem routines which extract the HDLC BPSK1000 frame. The BPSK1000 software modem routines then return the demodulated HDLC frames along with a length and a 32-bit CRC. These frames contain the telemetry values which ARISSatTLM then interprets and displays for the user.

Demodulated frames are also saved to a .CSV file on the local hard drive. The file will be named YYYY-MM-DD.CSV, where YYYY is the year, MM (or M) is the month, and DD (or D) is the date. Each

frame is saved to the file by appending the file if the file already exists. This ensures that no received telemetry is accidentally overwritten. The frames are written to the file with one line per frame. Each frame starts with the date YYYY/MM/DD, a hyphen, the time (UTC) as HH:MM:SS, then each (raw) byte of the demodulated frame is written as a decimal value. For example, a telemetry frame line starts as

```
2010/06/26-14:55:48,084,107,014,000,00,072,105
```

and an experiment frame starts as

```
2010/06/26-14:55:44,069,049,066,053,054,159,000
```

The .CSV file can then be imported into any number of common spreadsheet programs for further analysis.

ARISSat-1 Frames

There are a couple different types of telemetry frames from ARISSat. The experiment frames contain data from the Kursk experiment and the telemetry frames contain ARISSat telemetry.

ARISSAT-1 Experiment Frame

Please see Table 1.

The transmitted frames are expected to alternate between ARISSat telemetry frames and numbered experiment frames. The order should be T, E1B1, T, E1B2, T, E1B3, T, E1B4, T, E1B5. The Kursk experiment runs on a schedule and the experiment frames will most likely be the same five frames

during any given satellite pass unless the experiment has just completed. The ARISSat telemetry frames will each be unique. There will probably be less than a second of delay between frames, during which only HDLC flags will be transmitted, but it will take several seconds to send a complete telemetry frame. [6]

The ARISSat telemetry frames consist of a frame ID, a counter, a traditional satellite "Hi" message, a collection of telemetry bytes based on a C language structure named `ss_telem_t` and a CRC32. The actual definition of the `ss_telem_t` structure and all of the fields contained within it can be found at the bottom of the `ss_types_common.h` header file on the SVN in the "firmware common" directory. Interested readers may also want to consult the `ss_stdint.h` file in the same directory. The `ss_telem_t` structure within the ARISSat telemetry frame is a byte for byte copy of contents of a `ss_telem_t` struct in memory on the satellite.

ARISSAT-1 telemetry frame

Please see Table 2 (next page).

The `ss_telem_t` structure can be broken down into these major parts.

Please see Table 3 (next page).

Description	Size	Notes
frame ID byte "E"	1 byte	hex 0x45, ASCII letter "E" for experiment
Experiment number	1 byte	This will always be "1" for the Kursk experiment.
Experiment Block ID	1 byte	hex 0x42, ASCII letter "B"
Experiment Block number	1 byte	While still on the satellite, the experiment data is broken into five pieces for transmission. This byte is the piece number (1-5) as an ASCII character 1,2,3,4 or 5 (hex 0x31 to hex 0x35).
Mission Elapsed Time	4 bytes	Mission Elapsed Time when the IHU received this 2k of experiment data from the Kursk experiment.
Experiment data	varies	Experiment data is broken up into five pieces for transmission. Four pieces will be 512 bytes. The fifth piece will be 36 bytes. ARISSat will join the pieces together into a Kursk experiment data frame [4] and will save it as a file called KURSK-nnnnn.EXP where nnnnn is the Mission Elapsed Time when the Kursk experiment frame was received.

Table 1: ARISSat-1 Experiment Frame.



Description	Size	Notes
frame ID byte "T"	1 byte	hex 0x54, ASCII letter "T" for telemetry
frame counter	4 bytes	4 byte integer counter
"Hi" message	22 bytes	The ASCII string "Hi, This is ARISSat-1."
ss_telem_t structure	290 bytes*	the ss_telem_t structure from the ss_types_common.h file
CRC32	4 bytes	CRC-32 as calculated by the spacecraft. Even though the demodulator software will verify the CRC to confirm a valid frame, we retain the CRC for end-to-end integrity checking.[5]

Table 2: ARISSat-1 telemetry frame.

* The size of the ss_telem_t structure (and thus the total size of the "T" telemetry frames) has changed numerous times during the development of the satellite. It is possible that the size will change again before the satellite is launched. The information here should be considered to be preliminary. The ss_types_common.h file on the SVN will have the latest version of the ss_telem_t struct.

Description	Size	Notes
Mission time	4 bytes	Mission Elapsed time (in seconds). This counter can reset back to zero.
Mission mode	4 bytes	One of "Emergency Power", "Lower Power", or "High Power"
IHU Temperatures	18 bytes	Temperatures measured from the IHU
Power	75 bytes	Current consumption and ON/OFF status for different parts of the satellite
PSU Status	21 bytes	PSU statistics
PPT Status	18 bytes	Solar panel current, voltage, temperature, etc. There are six copies in the frame (one for each solar panel and PPT).
IHU Status	52 bytes	IHU, PSU, SDX, Camera ON/OFF status
Calc	8 bytes	Telemetry values calculated from raw counts by the satellite (voltage, current, temperature)

Table 3: Overview of the ss_telem_t structure in the ARISSatTLM frame.

Each of above telemetry sections can be broken down further into individual telemetry values. Unlike many other spacecraft which have single byte telemetry channels, ARISSat telemetry values vary in size from individual bit flags up through eight byte numeric values.

IHU Temperatures Telemetry

Please see Table 4. To convert the raw temperature values from these telemetry channels into degrees Celsius, the raw value should be scaled using the scale_thermistor_C() routine in the scale_therm.c file. This function will perform a lookup in a table that correctly converts the raw value to degrees Celsius.

Power Telemetry

Please see Table 5 (next page). The four two-bit camera status indicators will have a value of three for ON and zero for OFF. However, they will only have the value of three if the camera was actually turned ON

taking a picture at the exact instant when the telemetry frame was being created by the spacecraft. Because the cameras are being turned ON and OFF, the status bits might appear as OFF in the telemetry frames even when pictures are being taken.

PSU Status Telemetry

Please see Table 6 (next page). The PSU status telemetry is included for the ARISSat developers. These values are used for debugging purposes. ARISSatTLM does not currently display the PSU status telemetry.

PPT Status Telemetry

Please see Table 7 (page 30). There are

six copies of the PPT Status telemetry in each ARISSat telemetry frame since there are six solar panels and six PPTs on the spacecraft. In each PPT status section there are two bytes that are logically split in half such that each four bit nibble contributes to a different telemetry value. This allows four twelve bit telemetry values to be transmitted in six bytes.

IHU Status Telemetry

Please see Table 8 (page 30). The IHU status telemetry is also included for the ARISSat developers. These values are used for debugging purposes. ARISSatTLM does not currently display the IHU status telemetry.

Calculated Telemetry

Please see Table 9 (page 31). These values were calculated on-board the satellite and converted from raw counts to human readable formats.

Source Code Availability

For those who are interested in the ARISSatTLM source code, it is available on a software repository ("the SVN") along with over three gigabytes of other ARISSat-1 material. Under Windows, the SVN can be accessed using the (free) TortoiseSVN software [3], or directly via https. Interested readers can start at <https://svn.sarpeidon.net/viewvc>. The ARISSatTLM files are in the ground_station_software/ARISSatTLM directory. Currently the ARISSatTLM software can be compiled with gcc 4.2.4, in which case the executable requires the demod.dll and cygwin1.dll files, or with gcc 4.3.4, in which case the demod.dll, cygwin1.dll, and cyggcc_s-1.dll files are needed by the executable.

ARISSat Telemetry for Everyone

In addition to the Windows version of ARISSatTLM, there is also a version for

Description	Size	Notes
RF	2 bytes	RF module internal temperature
Control Panel	2 bytes	Control panel internal temperature
Experiment	2 bytes	Experiment temperature
Bottom Camera	2 bytes	Bottom Camera temperature
Top Camera	2 bytes	Top Camera temperature
IHU Enclosure	2 bytes	IHU Box temperature
Battery	2 bytes	Battery temperature
PSU PCB	2 bytes	PCB inside the IHU Box
IHU PCB	2 bytes	PCB inside the IHU Box

Table 4: IHU Temperatures Telemetry.



Description	Size	Notes
Battery Status	12 bytes	2 bytes raw voltage; 2 bytes raw current; 2 bytes 2.5V reference; 6 bytes flight total net coulombs into battery
Battery History	12 bytes	6 bytes total charging into battery; 6 bytes total discharge out of battery
Camera Power	9 bytes	6 bytes total raw coulombs; 12 bits raw current value; four 2 bit status indicators for the four cameras
Experiment Power	8 bytes	6 bytes total coulombs; 12 bits raw current; 2 bits status
IHU	8 bytes	6 bytes total coulombs; 12 bits raw current; 2 bits status
SDX	8 bytes	6 bytes total coulombs; 12 bits raw current; 2 bits status
Power Supply 5V	8 bytes	6 bytes total coulombs; 12 bits raw current; 2 bits status
Power Supply 8V	8 bytes	6 bytes total coulombs; 12 bits raw current; 2 bits status
RF	1 byte	RF Power on/off
Spare	1 byte	Spare power on/off

Table 5: Power Telemetry.

Apple computers (ARISSat TLM) that is being developed independently by Gilbert Mackall. We are attempting to keep the telemetry display pages as similar as possible between architectures. Both programs use the exact same DSP routines for demodulating the audio signal into the HDLC frames and on-the-air performance is expected to be identical.

Those with mobile devices are not going to be left out either. We have plans for a Web page on the www.arissattlm.org Web site which will contain (after ARISSat-1 is in orbit) a copy of the most recently received telemetry. There is a demonstration page at <http://www.arissattlm.org/mobile>. Currently this page is loaded with fictitious data, but it shows what the telemetry should look like when formatted for the smaller screens on mobile devices such as iPod touch and iPhone.

The Internet Telemetry Reflector

When the ARISSatTLM software is completed, demodulated frames will be sent to an Internet telemetry reflector. The reflector software will then combine all of the received telemetry frames, send telemetry frames in real time to anyone on the Internet who is interested using the Satellite Telemetry Protocol, update the www.arissattlm.org Web site with the latest telemetry, and update the telemetry archives.

Conclusion

Comments and criticisms on ARISSatTLM and the source code are welcome, as are bug reports and bug fixes. [7]

Screenshot - ARISSatTLM main window (Windows Version)

Please see Figure 1 (page 31). In this (earlier) version of ARISSatTLM the `ss_telem_t` struct was only 235 bytes. The `ss_telem_t` struct begins immediately after the ASCII "Hi" message. The Mission Elapsed Time starts at byte 0x01B in the hex dump. Note that the four bytes starting at byte 0x01B are "31 CC 02 00". As a four byte integer value

Description	Size	Notes
Oscillator status	1 byte	Oscillator status
Crystal failed count	1 byte	Crystal failed count
Pin disturbs	1 byte	Flags if pin settings need to be changed
IHU forced reset count	1 byte	IHU forced reset count
SDX automatic turn on count	1 byte	SDX automatic turn on count
IHU recoveries count	1 byte	IHU recoveries count
Camera shutdown count	1 byte	Shutdowns due to low batteries
I2C wcol errors	1 byte	For I2C debugging
I2C overflow errors	1 byte	For I2C debugging
I2C bad message count	1 byte	Counts messages not formatted correctly
I2C bus idle resets count	1 byte	Counts resets due to lack of activity
I2C bus hang resets count	1 byte	Counts resets due to a stuck message
I2C bus reset sspcon	1 byte	SSPCON before bus is reset
I2C bus reset sspstat	1 byte	SSPSTAT before bus is reset
Command mismatch count	1 byte	Commands that were repeated to execute
PPT Poll Busy count	1 byte	Time to start the next poll and PPT is still being polled
PSU Reset reason	1 byte	The reset reason
PSU self-reset count	1 byte	Count of the number of times the PSU has reset itself

Table 6: PSU Status Telemetry.

this becomes 0x0002CC31, which converted to decimal is the Mission Elapsed Time of 183345 seconds (compare with "Telemetry As Text" screenshot below). The mission mode is the next four bytes are 0x00000002, which corresponds to "High Power".

Screenshot for Windows ARISSatTLM Telemetry As Text. Compare with MAC ARISSatTLM screenshot.

Please see Figure 2 (page 31). The "^^" and "v" characters indicate increasing or decreasing values compared to the previously received telemetry frame.

Telemetry - Mobile Version

See Figure 3 (page 31). Also see: Screenshot - <http://www.arissattlm.org/mobile>

The layout for the telemetry values have been reformatted for display on smaller hand held devices.

Screenshot for Mac ARISSatTLM. Compare with Windows. ARISSatTLM Telemetry as Text screenshot.

Please see Figure 4 (page 31).

References

- [1] Monteiro, Anthony; "Space Radio for Windows"; *Proceedings of the AMSAT-NA 26th Space Symposium*. 2008.



Description	Size	Notes
Panel Energy	6 bytes	Solar panel energy since last reboot
Panel Voltage	1 byte	Solar panel voltage
Panel Current	2 bytes	Contains 10 bit analog-to-digital converter output of the instantaneous current from this solar panel
Panel Temperature	12 bits	Solar panel temperature
Diode Temperature	12 bits	Diode temperature
Inductor Temperature	12 bits	Inductor temperature
FET Temperature	12 bits	FET temperature
Setpoint	1 byte	Solar panel current PWM setpoint
Age	1 byte	Indicates the age of the information from this PPT. A zero value indicates current information.
Corrupt count	1 byte	Count of corrupt packets from this PPT

Table 7: PPT Status Telemetry.

- [2] Karn, Phil; “The BPSK1000 Telemetry Modem for ARISSat-1”; *Proceedings of the AMSAT-NA 28th Space Symposium. 2010.*
- [3] TortoiseSVN is available at <http://tortoisesvn.net/>
- [4] The format of the Kursk experiment data frame is documented on the SVN at: <https://svn.sarpeidon.net/viewvc/suitsat2/Experiments/Kursk%2011509/Suitsat%202/Structure%20of%20experiment%20data%20frame%2B.pdf>
- [5] Karn, Phil; “The Satellite Telemetry Protocol (STP)”; 8/1/2010; https://svn.sarpeidon.net/suitsat2/repos/ground_station_software/ARISSatTLM/Documentation/telem.txt
- [6] The task of calculating the exact time to transmit a telemetry frame given the frame size in bytes, factoring in the transmit bit rate, including the overhead for forward error correction, the CRC32, the two necessary HDLC flag bytes and any data dependent HDLC bit stuffing, is left as an exercise to the reader.
- [7] The author encourages constructive criticisms and particularly welcomes deconstructive comments and puns. ☺

Description	Size	Notes
Oscillator status	4 bytes	Oscillator status
Oscillator failed flag	1 byte	Flag if the IHU EC oscillator has failed
Core missed ticks counter	1 byte	Number of missed ticks in the core timer
Bad state flag	1 byte	Flag if illegal/unknown state is detected*
Bad NAK counter	1 byte	Unexpected NAKs*
Bus timeout counter	2 bytes	Bus timeouts*
Received too big	1 byte	Received too big
Receive errors	1 byte	Overrun / framing errors
UART receive errors	1 byte	Overrun / framing errors
Current state	1 byte	Current state
Bad state flag	1 byte	Illegal / unknown state detected
Full experiment runs count	1 byte	Number of times the experiment has been run fully
Quick experiment runs count	2 bytes	Number of times the experiment has been run just to collect data
Aborted power count	2 bytes	Number of times the experiment was running but was aborted due to power reasons
Quick timer	4 bytes	Timer for next experiment event
Full timer	4 bytes	Timer for next experiment full run
I2C timeout failed	1 byte	I2C timeout failed
Reset reason from RCON	4 bytes	Reset reason from RCON
Error PC	4 bytes	Error PC
PSU delayed response count	2 bytes	PSU request was still pending when the IHU went to process it
PSU error	2 bytes	PSU request contained an error when the IHU went to process it
PSU last error	1 byte	Last errored PSU request results
PSU stale seconds	2 bytes	How stale the PSU data is (in seconds)
ROM IDs sent counter	2 bytes	Number of ROM IDs sent
SD IDs sent counter	2 bytes	Number of SD IDs sent

Table 8: IHU Status Telemetry.

*** These values are repeated twice.**



Description	Size	Notes
Battery voltage	2 bytes	Over the range -327.68 to 327.67 volts
Battery current	2 bytes	Over the range -4096 to 4096 mA (125uA/lb)
RF (8V) current	2 bytes	Over the range -4096 to 4096 mA (125uA/lb)
IHU Enclosure temperature	1 byte	IHU Enclosure temperature (-128 to 128 degrees C)
Control panel temperature	1 byte	Control panel temperature in degrees C

Table 9: Calculated Telemetry.

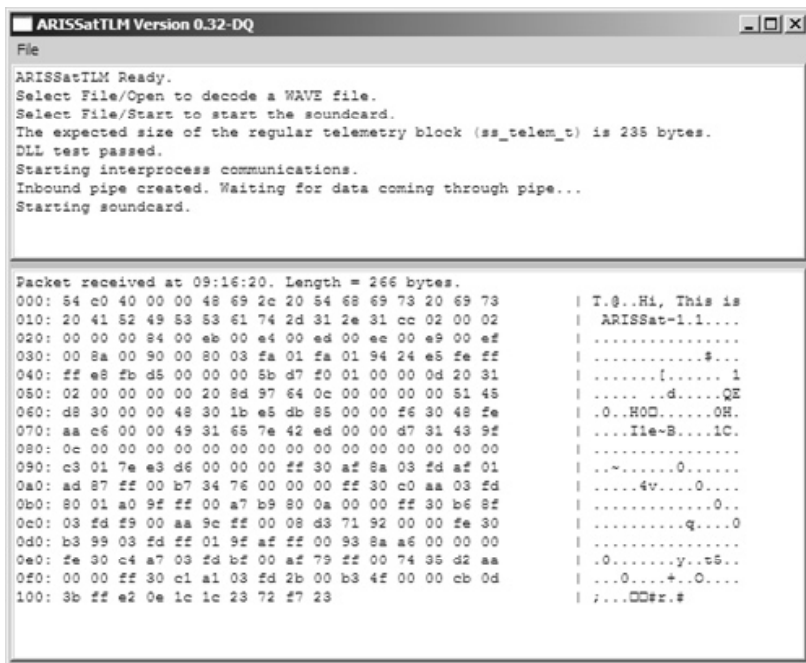


Figure 1: Windows Version Screenshot.

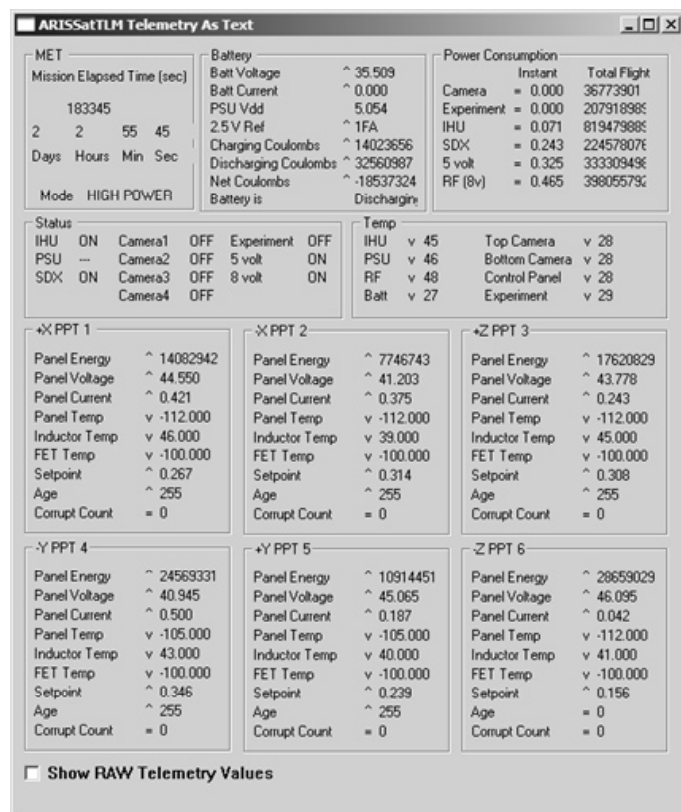


Figure 2: Windows Telemetry as text.

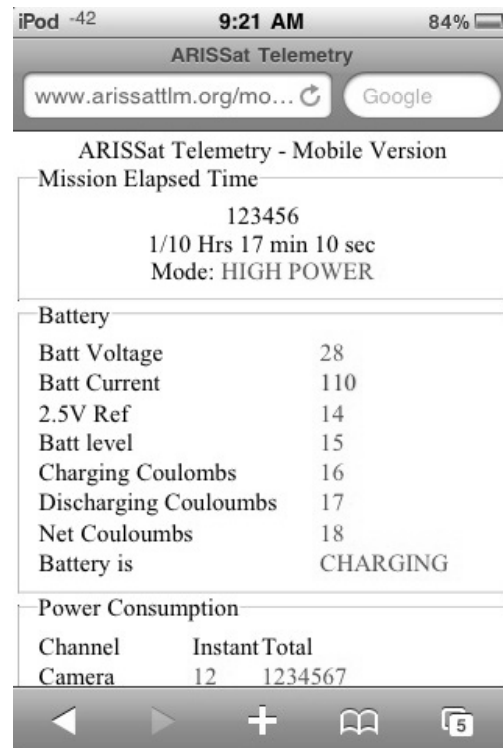


Figure 3: Telemetry - Mobile Version.

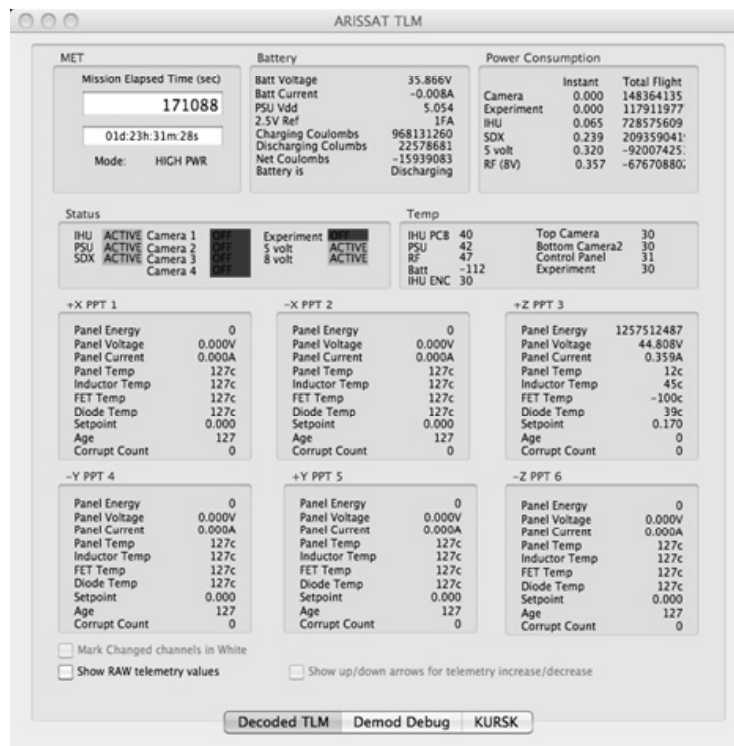


Figure 4: Mac Screenshot - ARISSatTLM Telemetry as Text.



Reach for the stars.

N3UC, formerly N4USI, control operator of AO-27, says, "AO-27 may be getting old and failing at times, after 15 years circling the earth. But, my IC-910H that I use to control AO-27, using data transmissions, has never failed me".

Icom 910H The perfect satellite rig.



Have fun communicating with these satellites using the IC-910H:

AO-27, SO-50, AO-51, and VO-52.

In the near future, check out these satellites:

AMSAT-Phase 3E, AMSAT-Eagle, SumbandilaSAT, and KiwiSAT.

Coming to the sky near you! Visit an authorized Icom dealer today!

